

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

***PROJETO DE UM SISTEMA OPERACIONAL
MULTIPROGRAMADO PARA ENSINO E
APLICAÇÕES DIDÁTICAS***

MARIA DA GRAÇA BRASIL ROCHA



São Carlos-SP

**PROJETO DE UM SISTEMA OPERACIONAL
MULTIPROGRAMADO PARA ENSINO E
APLICAÇÕES DIDÁTICAS**

MARIA DA GRAÇA BRASIL ROCHA

Orientador: Prof. Dr. João Antonio Zuffo

**Dissertação apresentada ao Instituto
de Ciências Matemáticas de São Car-
los, da Universidade de São Paulo,
para obtenção do título de Mestre em
Ciências de Computação e Estatística**

**- São Carlos -
1986**

Ao Irineu e aos meus pais

RESUMO

O objetivo primeiro deste trabalho é criar uma ferramenta de auxílio para o ensino de Sistemas Operacionais. Sua principal função é propiciar ao aluno uma experiência prática em cima dos conceitos teóricos que devem ser vistos, já que terá em mãos um Sistema Operacional que poderá ser modificado.

ABSTRACT

The first objective of this project is to create an aid tool for the operating systems teaching. Its main function is to propitiate to students a practical experience about the theoric concepts that must be seen, once it will be at hands an operating system that may be modified.

I N D I C E

I - INTRODUÇÃO	01
II - SISTEMAS OPERACIONAIS MULTIPROGRAMADOS	
2.1 - Evolução dos Sistemas e Tendências Atuais ...	04
2.2 - Exemplos de Sistemas Operacionais Multiprogramados	11
2.3 - Linguagens de Alto Nível para Programação de Sistemas Operacionais	16
III - PROJETO	
3.1 - Apresentação Global	20
3.2 - Simulador	
3.2.1 - Características Gerais	22
3.2.2 - Simulação do "Hardware"	26
3.3 - Núcleo	34
3.4 - Sistema Operacional	38
3.5 - Aspectos da Implementação	45
IV - RESULTADOS E CONCLUSÕES	47
BIBLIOGRAFIA	60
APÊNDICE I	
Repertório de Instruções	64
APÊNDICE II	
Um Exemplo	71
APÊNDICE III	
Especificação do Sistema	128

CAPÍTULO I

INTRODUÇÃO

Num sistema computacional podemos identificar quatro componentes: o "hardware", o Sistema Operacional, os programas aplicativos (editor de textos, compiladores, etc..) e os usuários.

O "hardware" fornece os recursos básicos de computação e, os programas aplicativos definem a utilização desses recursos para a solução de problemas pelos usuários.

Nesse contexto, o Sistema Operacional é um programa que atua como interface entre o usuário do computador e seu "hardware". Em outras palavras, é o responsável por fornecer um ambiente no qual o usuário possa executar seus programas.

Além disso, um sistema computacional possui vários recursos que podem ser requeridos para solucionar um problema: tempo de UCP, espaço em memória, armazenamento de arquivos, dispositivos de E/S, etc.. O Sistema Operacional atua como gerente desses recursos, de forma que os usuários possam utilizá-los da maneira mais eficiente possível, estabelecendo uma cooperação entre eles e policiando a forma pelo qual são atendidos.

O Sistema Operacional que nos propusemos desenvolver é um sistema que suporta multiprogramação, isto é, permite que mais de um programa de usuário esteja sendo executado em um determinado instante de tempo.

Para permitir multiprogramação, o sistema operacional deve compartilhar os recursos do sistema entre os programas em execução. No nosso caso, o sistema é composto de um equipamento para entrada de dados, uma unidade central de processamento, uma impressora e um disco. Alguns desses recursos, como por exemplo, o acesso ao disco, é alocado a um programa até que ele termine sua execução ou não necessite mais do recurso. Outros, como a UCP, são compartilhados dinamicamente, de forma que dá ao usuário a aparência de que cada um tem seu próprio recurso. No caso da UCP, o sistema operacional reserva um intervalo de tempo para rodar um programa, findo o qual, passa-a a outro e assim por

diante. Esse tempo é chamado de segmento de tempo ("time slice") e dá ao usuário a aparência de que cada programa possui sua própria UCP, que é igual a UCP física só que um pouco mais lenta.

Os detalhes do funcionamento de um sistema operacional multiprogramado serão vistos no capítulo dois.

A metodologia de desenvolvimento utilizada envolveu a estruturação do projeto do sistema operacional como um conjunto de processos escritos em linguagem de alto nível, apoiados em um núcleo básico escrito em linguagem montadora ("Assembler").

Os processos desse conjunto são concorrentes, ou seja, competem por recursos e dados compartilhados e cooperam entre si, executando tarefas parciais pertencentes a um determinado sistema. Tanto a competição quanto a cooperação necessitam de algum mecanismo para a sincronização e comunicação entre os processos.

Existem vários desses mecanismos para processos concorrentes, Segre [23], dá uma descrição completa dos mesmos. No capítulo 2, damos uma descrição detalhada de monitores [8], que foi o mecanismo que utilizamos nesse trabalho.

De acordo com a metodologia adotada, o núcleo é o responsável por multiplexar a UCP entre os processos concorrentes e de dar aos mesmos acesso exclusivo aos monitores. O núcleo do nosso sistema está descrito no capítulo 3.

Como a finalidade do trabalho é para utilização em ensino, o sistema operacional desenvolvido roda numa máquina simulada, o que além de facilitar a visualização da arquitetura implementada, permite que o aluno possa alterá-la facilmente. O simulador está descrito detalhadamente também no capítulo 3.

As mesmas facilidades ocorrem com o sistema operacional. A apresentação de cada processo que o compõe, seguida de alternativas de implementação, permitirá ao aluno conhecer os detalhes de forma a tornar-se capaz de alterá-lo ou até substituí-lo totalmente, podendo estudar o desempenho em cada caso e as dificuldades e consequências que tais alterações podem gerar com relação ao sistema como um todo. No capítulo 4, damos uma breve descrição de como isso pode ser feito.

Esse trabalho foi baseado na experiência feita no Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, descrita em Kirner [14] e no trabalho feito por Holt [11].

Tanto o sistema operacional quanto o núcleo foram escritos em Pascal Concorrente, mas por não dispormos do compilador dessa linguagem, a implantação foi feita em Pascal "Sequencial", o que exigiu algumas adaptações, de forma a permitir a concorrência entre os processos. Essas adaptações são descritas também no capítulo 3.

No apêndice 1 temos a descrição do repertório de instruções da máquina simulada.

No apêndice 2 apresentamos a listagem obtida da execução de vários programas.

No apêndice 3 está a especificação do sistema operacional em Pascal Concorrente.

O Sistema Operacional foi implementado num computador IBM/370 disponível na Divisão de Processamento de Dados da Universidade Federal de São Carlos.

CAPÍTULO II

SISTEMAS OPERACIONAIS MULTIPROGRAMADOS

2.1 - Evolução dos Sistemas e Tendências Atuais

Nos primeiros tempos da computação, costumava-se programar com instruções codificadas em binário que chamaremos de instruções numéricas, ou seja, em linguagem de máquina que utilizavam endereços absolutos e mapas de memória. O programa era perfurado em cartão ou fita de papel para ser carregado na memória, através de um outro programa o Disparador ("boot strap loader"). O programador podia monitorar a execução de seu programa com auxílio das chaves e luzes da console para a depuração dos erros.

O primeiro passo para a automação do uso dos computadores, foi o desenvolvimento de um programa montador, que eliminou a programação com instruções numéricas, já que sua tarefa era traduzir os códigos de operação das instruções que passaram a ser escritas em mnemônicos. Estas instruções eram passadas para cartões perfurados e, então carregadas e executadas. A consequência direta disso, foi facilitar a forma de escrever, alterar e executar os programas.

O passo seguinte, foi o surgimento do carregador relocável ("relocatable loader"), criando a facilidade de endereçamento simbólico e relocável. Com isso, o programador deixou de se preocupar com endereços absolutos de partes do programa, já que uma vez traduzido, ele podia ser carregado em qualquer parte da memória. A consequência imediata do carregador relocável foi o aparecimento de estruturas de programa como as subrotinas e bibliotecas.

Nesse ponto, tínhamos em fita magnética, a biblioteca de programas, os tradutores e programas de serviço a fim de tornar a carga desses programas rápida e fácil. Procurou-se também aperfeiçoar um conjunto de subrotinas que efetuassem de maneira padronizada as operações de E/S dos periféricos, um problema difícil naquela época. Tais módulos, chamados IOP ("Input,

Output Packages"), faziam parte da biblioteca de programas e eram chamados tanto pelos tradutores e programas de serviço como pelos programas de aplicação.

Com todos esses melhoramentos, tornou-se clara a ineficiência em relação ao tempo em que a máquina efetivamente funcionava para o tempo em que o programador gastava em preparar a execução, montar as fitas magnéticas respectivas, e pensar nas correções que deveria fazer no programa.

Por esse período o programador deixou de operar diretamente a máquina, surgindo a figura do operador profissional. Logo que um programa terminava, o operador podia iniciar o próximo e como ele era especializado na tarefa de montagem de fitas, o tempo gasto nessa tarefa foi reduzido. O usuário fornecia o que era necessário para rodar seu programa, como também uma pequena descrição de como ele rodava. Naturalmente, o operador não corrigia um programa pela console, já que ele não o entendia. Se o programa não estivesse correto, era fornecido ao seu dono uma descarga ("dump") de memória e registradores, permitindo assim que o operador providenciasse imediatamente a execução do próximo programa, deixando a tarefa de correção para o usuário.

Para acelerar mais ainda a execução dos programas, estes passaram a ser agrupados em lotes, de acordo com suas necessidades. Por exemplo, todos os programas escritos em Fortran eram agrupados para rodarem em sequência, economizando então, o trabalho envolvido na carga da fita com o compilador, com o mesmo procedimento adotado para quaisquer outros compiladores. Apesar de tudo, o operador ainda era o responsável por cuidar da transição de um programa para outro, já que quando um programa parava, ele devia observar a causa do término, para providenciar uma descarga ("dump") no caso de parada anormal, depois, carregar a leitora com o próximo programa e iniciar o computador. Durante essa transição a UCP permanecia ociosa.

Para eliminar esse tempo ocioso, desenvolveu-se o Sistema Monitor, que na sua forma mais simples consiste de um programa supervisor residente na memória principal e que através de uma série de comandos lidos de cartões de controle intercalados no meio da massa de cartões dos usuários, executa funções

de rotina como carregar um compilador ou um programa de serviço, executar módulo objeto recém-criado, etc., surgindo então o conceito de linguagem de controle de tarefas ("JCL"). Nesse ponto, o trabalho do operador ficou limitado a, alimentar a leitora de cartões com os programas e, recolher as listagens na impressora, com o tempo de transição entre os programas e carga reduzindo-se praticamente à zero.

Um sistema monitor típico consistia basicamente de:

- 1 - Uma rotina de E/S para ler os cartões de controle, e carregar os componentes do sistema monitor correspondentes;
- 2 - Uma rotina para interpretar os cartões de controle e chamar os tradutores e programas de serviço;
- 3 - Rotinas para decidir sobre condições e erros.

Nessa situação, cada vez que um programa ia ser executado, ele tinha domínio completo sobre todos os recursos do sistema, mesmo que não os usasse.

Com esses melhoramentos, novos esforços passaram a ser concentrados em tornar o sistema mais eficiente com respeito aos seus periféricos, já que eles operam em velocidade relativamente baixa, desenvolvendo-se o conceito de processamento em lotes ("Batch Processing").

Um lote de programas era armazenado em fita magnética com auxílio de um dispositivo independente. Essa fita magnética era usada como arquivo de entrada do sistema, no lugar da leitora de cartões. O sistema monitor passava então a leitura e processamento dos programas a uma velocidade maior, já que lia os programas na fita magnética. Da mesma forma, as saídas impressas geradas pelos programas em execução eram gravadas também em fita, e mais tarde, a impressão era feita a partir dessa fita com auxílio de outro dispositivo independente. Pela coordenação dessas atividades, era possível manter o computador sempre ocupado com lotes pré-gravados enquanto se fazia em paralelo a gravação de novos lotes e a impressão dos lotes já processados.

Apesar dessa evolução, os sistemas desse tipo ainda eram ineficientes por causa do tipo de processamento estritamente sequencial, o programa do usuário era lido, executado e produzia sua saída, e só após todos esses passos o próximo

programa podia ser iniciado, Além disso, se, no lote, tivéssemos um programa muito longo, ele atrasava o processamento dos outros programas do lote.

A introdução de interrupções e canais autônomos de E/S, possibilitaram o aumento na eficiência de processamento, já que permitem que as operações de E/S sejam feitas em paralelo com as atividades da UCP.

O sistema monitor de lotes evoluiu para a idéia de varredura cíclica ("spool"), que retira os dispositivos independentes para montagem e impressão das fitas magnéticas ficando esse trabalho para o próprio processador. Com a substituição das fitas magnéticas por tambores e discos magnéticos, já se podia executar os programas em ordem diferente da que eram lidos, uma vez que, esses dispositivos não tem o caráter sequencial das fitas, e surgia o mecanismo de agendamento de programas ("Scheduling"). Utilizando-se esse mecanismo, era possível minimizar o problema de atrasos no tempo de resposta provocado por um programa longo, agendando para execução primeiro os programas mais curtos.

A introdução dessas novas técnicas permitiu o surgimento de multiprogramação, isto é, a colocação de mais de um programa residente na memória principal, e cada um deles, tendo a oportunidade de prosseguir sua execução, durante os intervalos de espera pelo término da execução de E/S de outros. Logicamente, nesse tipo de ambiente, devemos ter um gerenciamento de memória. Como a memória tem tamanho limitado, não é possível acomodar inteiramente vários programas e naturalmente surgia o conceito de memória virtual paginada por "hardware".

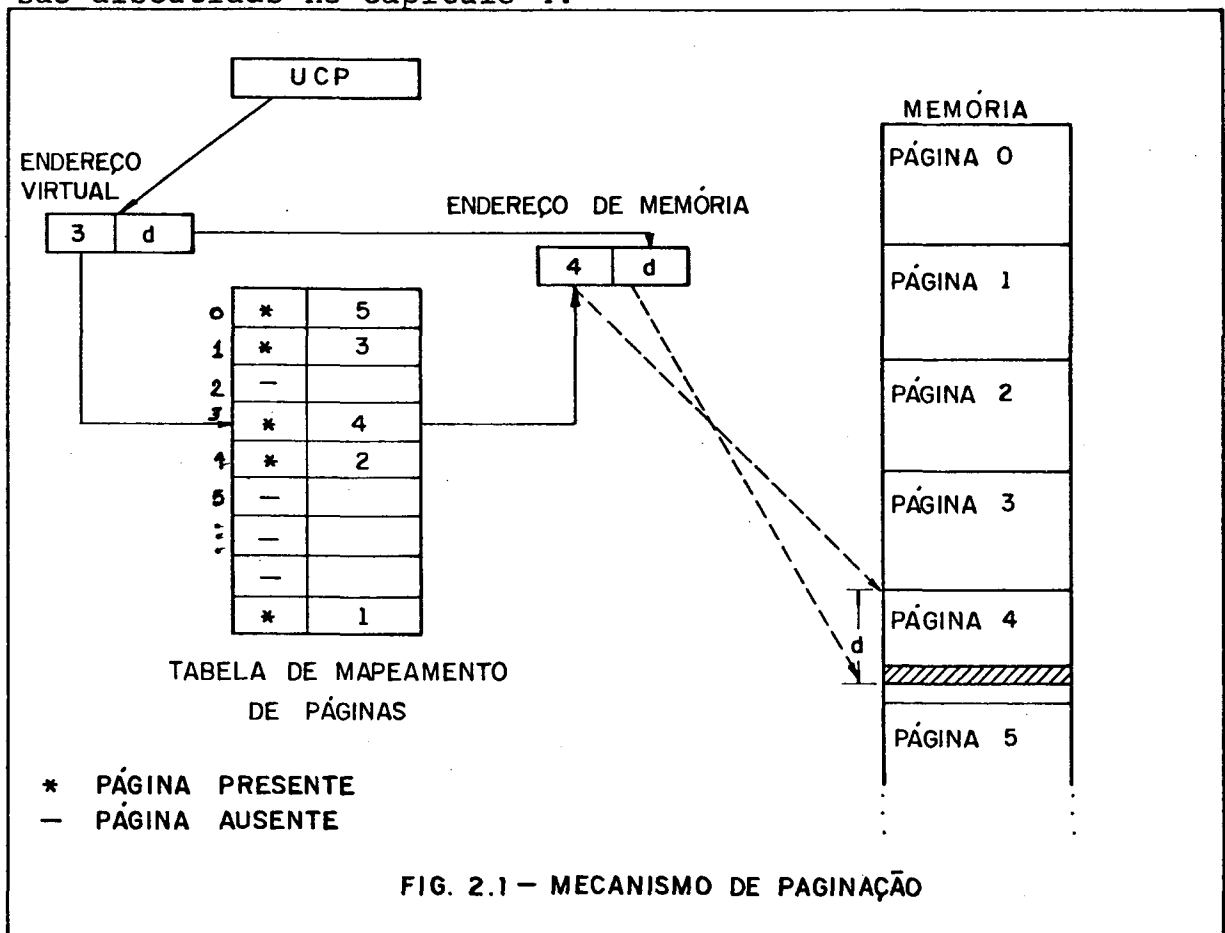
Nesse tipo de implementação, o programador escreve seu programa sem pensar no tamanho da memória principal, em princípio, basta que caiba no disco. Essa memória é dividida em blocos de tamanho físico determinado, que chamamos de páginas. Os programas dos usuários são lidos e colocados no disco, sendo que o espaço no disco também é dividido em páginas, com o mesmo tamanho das páginas da memória. O sistema operacional carrega algumas páginas do programa na memória e através de um mecanismo, associa o endereço da página no disco, com o endereço da mesma memória. Esse mecanismo é conhecido como mecanismo de

transcodificação de endereços e utiliza tabela de páginas.

Quando a UCP gera um endereço, ele é dividido em duas partes: o número da página, e o deslocamento dentro dela. O número da página é usado como entrada para a tabela de páginas e é verificada sua presença na memória. Se a página em questão já está carregada na memória, o endereço da mesma nesse dispositivo é obtido e, junto com o deslocamento dado no endereço, é obtido o endereço real. Esse mecanismo está ilustrado na figura 2.1.

No caso de ser determinada, a ausência da página endereçada na memória, verifica-se a existência de página não utilizada nesse dispositivo. Se houver, é feita uma cópia da página do disco para a memória, com a respectiva atualização da tabela de páginas, caso contrário, através de alguma política de substituição de páginas, escolhe-se uma para ser substituída pela página em questão. Enquanto é providenciada a substituição da página, um outro programa é ativado e colocado em execução.

As diferentes políticas de substituição de páginas são discutidas no capítulo 4.



O sistema Atlas [20], implementado em Manchester, Inglaterra, em 1961, utilizava esse esquema de memória virtual e a maior parte dos sistemas modernos que implementam memória virtual, utilizam variações desse mecanismo.

Existem várias formas de implementação da tabela de páginas. É possível construí-la através de uma memória associativa, que permite a localização do endereço físico de uma forma bastante rápida. Entretanto, por restrições de dimensão e, pelo custo de tal dispositivo, pode-se implementar uma tabela de páginas para cada programa na própria memória, e, o endereço dela, passa a fazer parte do contexto do programa, com o endereço físico obtido da forma já descrita.

A característica principal dos sistemas multiprogramados em lotes, é a não interação do usuário com seu programa enquanto ele está executando. Neste ponto, todos os esforços para aumentar a produtividade da máquina estão contemplados e, a preocupação passou a ser com a produtividade do programador.

O usuário submete seu programa para execução e, após um certo tempo recebe sua resposta. Como esse tempo de espera pode ser grande, qualquer eventual erro, leva a uma nova execução do programa e, uma nova espera. Para eliminar esse problema, surgiram então, os sistemas de tempo partilhado ("Time-sharing"). Nesses sistemas, o usuário trabalha num terminal de E/S do tipo vídeo ou teletipo, e pode interagir com o sistema em cada fase de execução de sua tarefa. Cada comando do usuário, é interpretado e executado imediatamente e em caso de ocorrência de erro, ele é corrigido antes de iniciar o próximo.

Normalmente, os programas são preparados no próprio terminal, via um editor de texto, depois são compilados, e no caso de erros, corrigidos via revisor e, só após livres de erros, submetidos para execução. Como a saída pode ser feita também via terminal, quaisquer resultados são imediatos. Quando satisfeito, o usuário pode então comandar a saída do seu programa via impressora.

Agora temos, portanto, a combinação da computação interativa com multiprogramação. Cada terminal ligado corresponde a um programa carregado na memória no sistema multiprogramado,

cada um ocupando áreas separadas de memória. A UCP é trocada rapidamente de um usuário para o outro, de forma a interpretar os comandos teclados em cada terminal, dando a impressão de que cada um tem sua própria UCP.

No sistema multiprogramado em lotes, procura-se compartilhar a UCP entre os vários usuários, através de tempo, de forma que nenhum deles espere indefinidamente, já no sistema em tempo partilhado, procura-se dar respostas imediatas a comandos teclados.

Outro tipo possível de sistema, é conhecido como Sistema em Tempo Real. Esse, é geralmente usado para controlar dispositivos em aplicações dedicadas. O computador recebe os dados através de sensores, analisa-os e se necessário ajusta controles. Sistemas que controlam experimentos científicos, sistemas de aplicações em medicina e controle industrial, são sistemas em tempo real, já que trabalham com tempos de respostas fixados no sentido de que se um processo requer atenção, o sistema deve responder dentro de um intervalo de tempo pré-fixado, uma vez que pode ocorrer perda de informação ou operação incorreta e fatal do mesmo.

Com esse tipo de sistema é usado para controlar vários processos simultaneamente, ele deve ter os mesmos mecanismos de compartilhamento de recursos do sistema de tempo partilhado, só que normalmente, os tempos de execução são conhecidos e previstos, ao contrário do tempo partilhado, em que as requisições dos recursos por parte dos usuários variam demasiadamente.

A classificação dos sistemas operacionais dada até aqui, tem em comum o fato de termos sempre tratado de máquinas que utilizam uma única UCP, mas a evolução dos conceitos de "hardware", levaram a projetos de sistemas com múltiplos processadores, ou seja, mais do que uma UCP, aumentando portanto o poder de computação.

Esse tipo de sistema, aparece normalmente em duas versões. Na primeira temos vários processadores compartilhando memória e periféricos, a cada processador é atribuída uma tarefa específica. Essa versão também é conhecida como Sistema Multi-processador.

A segunda versão, é o que conhecemos como rede de computadores. Em uma rede, computadores autônomos podem comunicar-se para trocar arquivos e informações entre si, entretanto, cada um tem seu próprio sistema operacional e trabalham independentemente.

Esse pequeno histórico da evolução dos sistemas, foi escrito utilizando-se várias referências. Entre elas podemos destacar [6], [16], [20], [22], [29].

2.2.- Exemplos de Sistemas Operacionais Multiprogramados

Vários sistemas operacionais foram construídos utilizando o conceito de multiprogramação, dessa forma, antes de detalharmos o nosso, vamos descrever alguns deles.

Inicialmente, vamos introduzir o conceito de monitor, que é o mecanismo de sincronização utilizado por vários desses sistemas, inclusive o nosso.

Chamamos de região crítica, o trecho de programa que tem acesso a um recurso compartilhado. A finalidade do monitor é garantir que regiões críticas associadas a um recurso se excluam mutuamente no tempo. Com isso, um processo só executa a região crítica se não existir nenhum outro processo na mesma.

O monitor agrupa num mesmo módulo todas as regiões críticas associadas a um mesmo recurso. Cada região corresponde a um procedimento que pode ser chamado pelos processos. A execução de um procedimento do monitor por um processo, impede a execução de qualquer outro procedimento do mesmo monitor por outro processo.

O monitor define uma estrutura de dados compartilhada e todas as operações que os processos podem executar sobre ela. É definida também a operação inicial que deve ser executada quando a estrutura de dados é criada.

Os processos só atuam sobre os dados compartilhados chamando procedimentos do monitor. Quando um processo executa um procedimento do monitor, ele tem acesso exclusivo aos dados compartilhados. Se um processo entra num monitor e não encontra

a condição necessária para prosseguir, como por exemplo, a disponibilidade de um determinado recurso, ele deixa o monitor e é colocado numa fila até que a condição seja satisfeita. Mais tarde, quando outro processo entrar no monitor e verificar que a condição acima é satisfeita, ele sinaliza a fila correspondente a condição, deixa o monitor, que ativa o processo que estava esperando para prosseguir a execução.

Esse é o conceito geral de monitor proposto por Hoare [8]. Existem variações desse modelo proposta por outros autores e, são discutidas em Segre [24]. O modelo que adotamos foi o proposto por Brinch Hansen [3] que é definido pela linguagem Pascal Concorrente descrita na segunda parte desse capítulo.

O sistema operacional SOLO, desenvolvido por Brinch Hansen [3], foi um dos que tiveram grande influência no nosso projeto e também em vários outros como veremos. O SOLO, suporta edição, compilação e armazenamento de programas escritos em Pascal. Esses programas podem acessar console, leitora de cartões, fitas e discos. A entrada, processamento e saída, são manipuladas por processos concorrentes. O sistema foi escrito em Pascal Concorrente, com menos de 4% escrito em linguagem de máquina.

O SOLO é constituído por módulos de programas como mostra a fig. 2.2, cada um deles é responsável pelo controle de um particular recurso do computador e, de um conjunto de processos concorrentes que utilizam esses recursos.

Os módulos são constituídos pelos seguintes elementos:

- 1 - Gerente de recursos, que controla a escalação das requisições da console e do disco entre os processos concorrentes;
- 2 - Gerente de console, que permite a comunicação de um processo com operador, após ele ter ganho acesso à console;
- 3 - Gerente de disco, permite aos processos terem acesso a arquivos em disco e seu diretório;
- 4 - Gerente de programas, busca programa no disco e o carrega na memória sob demanda dos processos que desejam executá-lo;
- 5 - Gerente de acopladores ("buffers"), transmite dados entre processos.

O processo que executa programas trabalha a partir de requisições do operador. O processo de entrada é quem produz os

dados para o processo que executa programas, já o processo de saída consome os dados produzidos pelo executor. O processo de carregamento, faz a carga e reinicia o sistema operacional quando o operador aperta a chave "liga" na console, o processo inicial inicia o resto do sistema após o sistema operacional ter sido carregado.

Cada um dos módulos de programa, é constituído de um ou mais tipos abstratos de dados (monitores e classes).

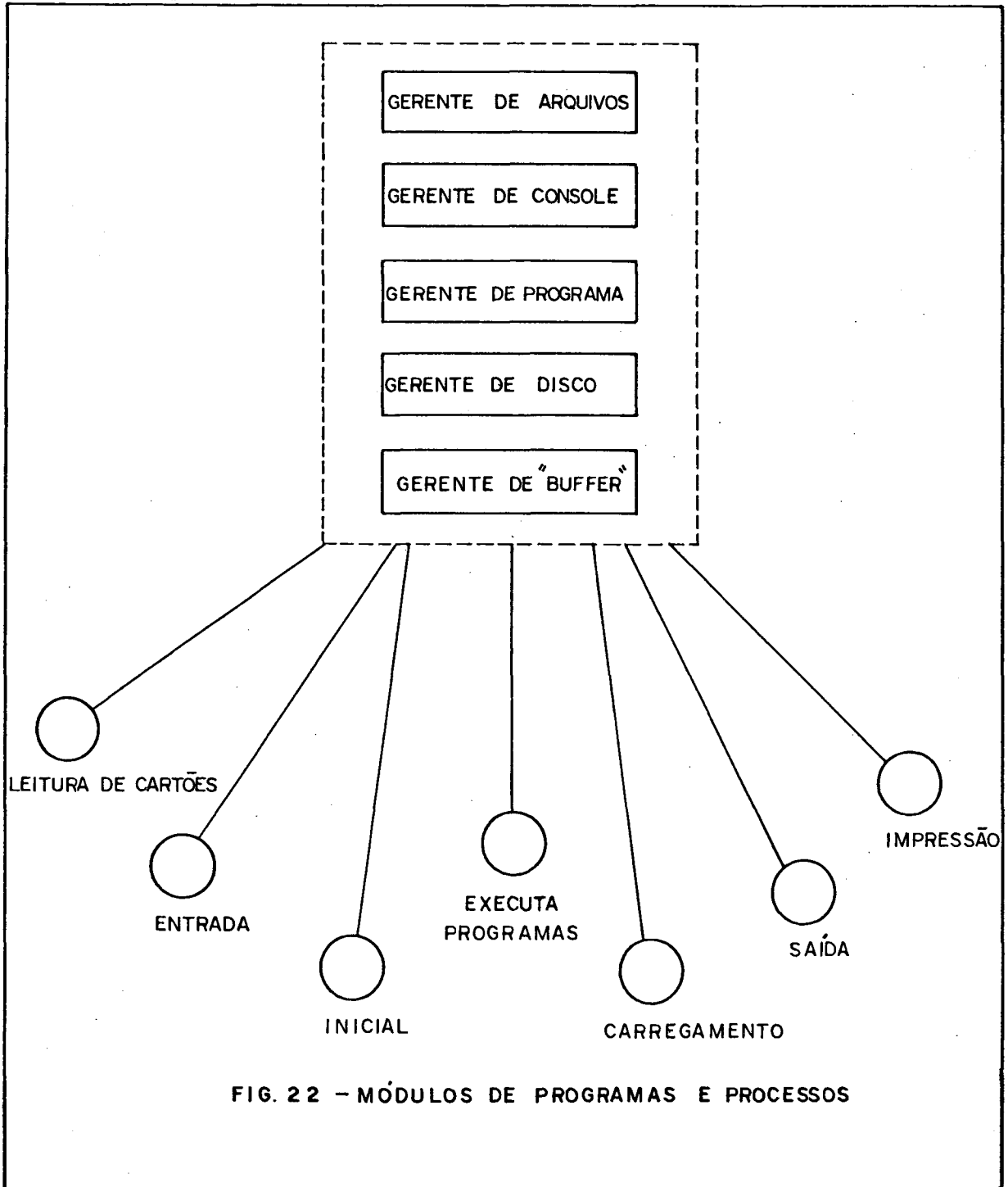


FIG. 22 - MÓDULOS DE PROGRAMAS E PROCESSOS

O núcleo do sistema SOLO, foi escrito em linguagem Montadora ("Assembler") e, é o responsável por multiplexar a UCP entre os processos concorrentes do sistema, e de dar a eles acesso exclusivo aos monitores.

Outro sistema que contribuiu para o nosso projeto, foi o sistema operacional desenvolvido na Universidade de Toronto por Holt [11]. Esse sistema funciona sobre um "hardware" simulado, que tem uma UCP, memória de usuário, memória de controle para o sistema operacional, uma leitora de cartões, uma impressora, um disco e uma console.

A UCP é simulada por um interpretador para a linguagem do usuário, que é uma linguagem de máquina bastante simples. A linguagem usada para escrever o sistema foi a CSP/K.

O sistema é dividido em três grandes componentes:

- 1 - Varredura Cíclica ("Spool") de entrada, que lê os programas dos usuários e carrega-os no disco;
- 2 - Executivo que gerencia a execução dos programas;
- 3 - Varredura Cíclica ("Spool") de saída, que pega os programas já executados no disco e os imprime.

O esquema desse sistema é melhor visualizado pela figura 2.3.

O núcleo CSP/K desse sistema operacional compartilha a UCP entre os vários processos. Como a linguagem de implementação suporta monitor, a comunicação entre processos se dá através desse mecanismo.

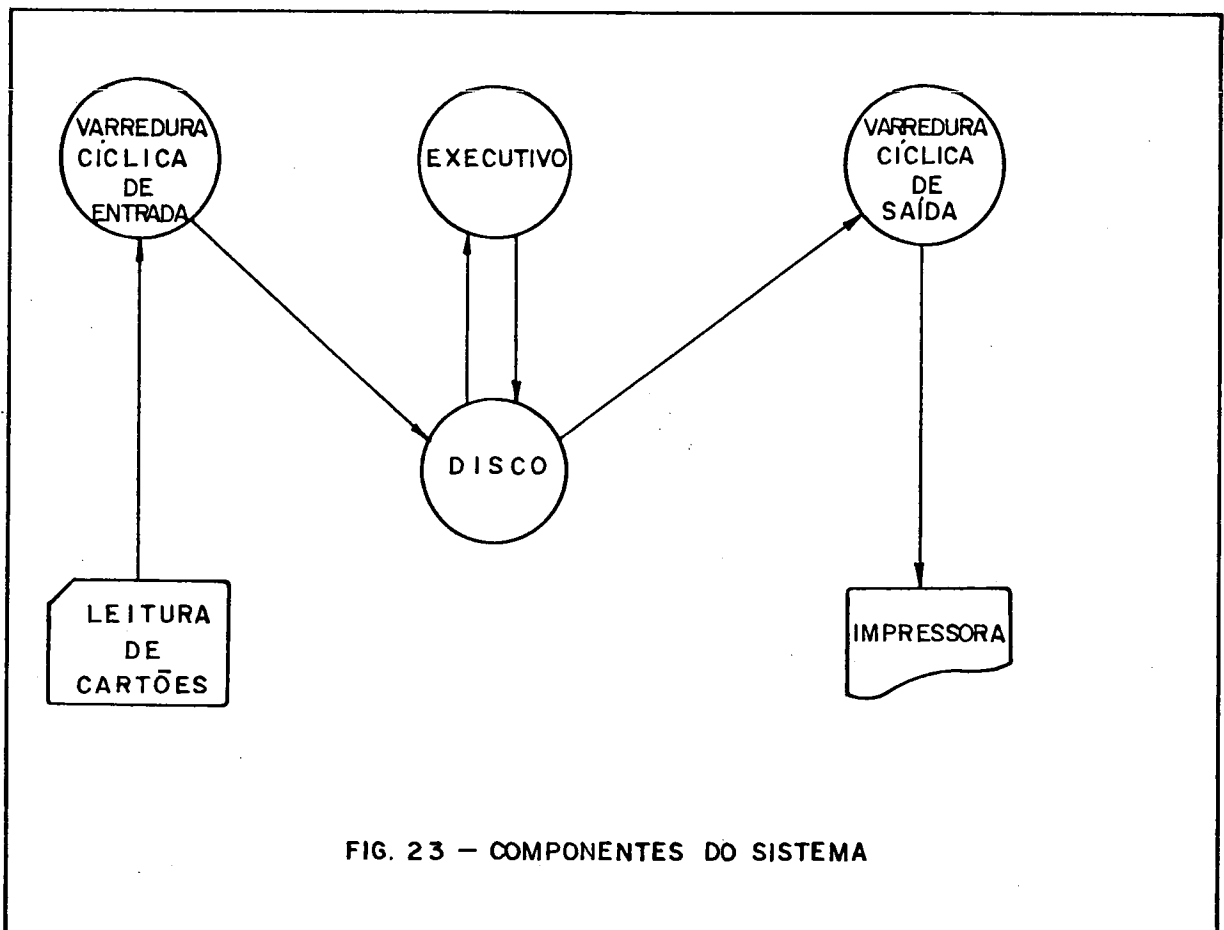


FIG. 23 - COMPONENTES DO SISTEMA

A partir da experiência com o sistema SOLO, Brinch Hansen desenvolveu o sistema TRIO [4], que permite o desenvolvimento e execução simultânea de programas em três terminais. O sistema operacional foi escrito em Pascal Concorrente e foi implementado em um minicomputador PDP 11/45. O programa concorrente TRIO, reside permanentemente na memória principal, junto com os três processos dos usuários de tamanho fixo. Cada processo de usuário está relacionado com um terminal. Os processos dos usuários, executam um programa Pascal cíclico chamado "DO", que aceita comandos do seu terminal correspondente. Nesse sistema, os usuários podem editar, compilar e executar programas escritos em Pascal.

Aproveitando-se também do sistema SOLO, Kruijer [15], desenvolveu um sistema operacional chamado Multi, só que voltado para aplicações comerciais. O sistema foi escrito em Pascal Concorrente, utilizando o SOLO e seus utilitários. O Multi,

suporta até quatro usuários via terminal, desenvolvendo programas em Pascal. O núcleo usado no sistema Multi, é uma adaptação do usado pelo SOLO.

Com a crescente popularidade dos microcomputadores, vemos também o desenvolvimento de sistemas operacionais multiusuários para os mesmos, já que tradicionalmente são usados com sistemas mono-usuários. Nessa linha, temos o sistema RIO, escrito por Smith [26], que foi implementado em um microprocessador Zilog Z80. O RIO foi desenvolvido para suportar um número fixo de processos, que no caso é 16, e a comunicação entre os mesmos se dá através de semáforos. Neste sistema, o tratamento de interrupção não é implementado através do núcleo, e sim pelo próprio mecanismo de interrupção do Zilog Z80. O sistema todo foi escrito em PLZ/SYS, que é uma linguagem de alto nível baseada no Pascal.

Ainda, no mesmo sentido, McKendry [18], desenvolveu o sistema Microps, para o microprocessador Motorola 6800. Foi programado na linguagem MOHLL ("Machine Oriented High Level Language") e a sincronização e comunicação entre processos é feita através de monitores.

Como existe um número muito grande de sistemas operacionais desenvolvidos a partir do conceito de multiprogramação, acreditamos que aqueles que aqui foram citados, servem de referência para situarmos o desenvolvimento da área.

2.3. - Linguagens de Alto Nível para Programação de Sistemas Operacionais.

Todos os sistemas citados na seção anterior, foram escritos em linguagem de alto nível, mas, anteriormente a eles, os sistemas operacionais eram escritos em linguagem Montadora. Isso ocorria porque as linguagens de alto nível não permitiam que escrevessemos códigos dependentes da máquina, como por exemplo, os acionadores ("Drivers") de E/S e, também não tinham mecanismos adequados para escrever programas concorrentes.

Começaram então a surgir as linguagens de alto nível

que suportam concorrência. Como é fácil perceber, o uso de tais linguagens é mais vantajoso, já que é muito mais simples escrever programas, corrigi-los, modificá-los e, além disso, propicia a portabilidade. A principal desvantagem apontada é a ineficiência de código gerado, ficando portanto como responsabilidade para os projetistas de linguagem, a otimização dos compiladores.

Como na seção anterior nos referimos às linguagens com as quais os sistemas foram desenvolvidos, vamos falar um pouco sobre o tipo de estrutura de dados que elas fornecem e, como não podia deixar de ser, vamos nos ater particularmente aos detalhes do Pascal Concorrente, que foi a linguagem por nós utilizada.

A linguagem Pascal Concorrente foi desenvolvida por Brinch Hansen [3] e, baseada no Pascal desenvolvido por Wirth [27]. A programação de sistemas é feita, a partir de três módulos básicos: processos, monitores e classes.

Um processo consiste de uma estrutura de dados, e um programa sequencial que opera sobre ela, como mostra o esquema na fig. 2.4. Os processos tem comportamento independente do tempo somente se suas variáveis são inacessíveis a outros processos. Se um processo usa valores de variáveis que outros processos também tenham acesso, o resultado vai depender da velocidade relativa dos outros processos.

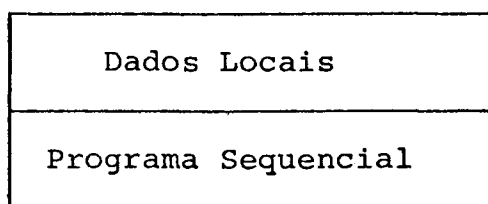


FIG. 2.4 - UM PROCESSO

O monitor é o mecanismo pelo qual é garantida a sincronização entre processos concorrentes e a transmissão de dados entre eles. Ele também controla a ordem na qual os processos usam os recursos compartilhados.

Um monitor define uma estrutura de dados compartilhada e todas as operações que os processos podem realizar com

esses dados. Essas operações são chamadas de procedimentos do monitor. Também é definida uma operação inicial que é executada quando suas estruturas de dados são criadas.

Na figura 2.5 abaixo, temos o esquema do monitor.

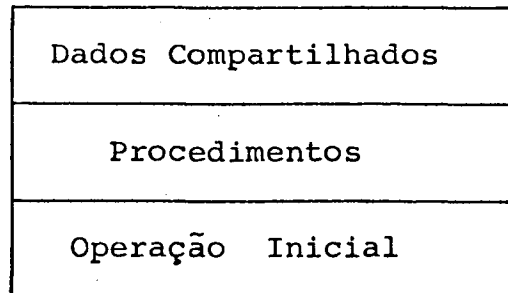


FIG. 2.5 - UM MONITOR

Os processos não podem operar diretamente os dados compartilhados, eles só podem chamar procedimentos do monitor, que têm acesso aos dados, cuja execução ocorre com exclusão mútua. Um procedimento do monitor é executado como parte de uma chamada de um processo e, nesse caso, o processo tem acesso exclusivo aos dados compartilhados enquanto executar o procedimento.

O Pascal Concorrente tem um tipo de dado chamado "queue", com capacidade para armazenar a configuração de um processo, que é usada por procedimentos do monitor para fazer o controle da escalação dos processos. Dessa forma, o monitor pode atrasar um processo na "queue" ou continuar um processo que estava esperando na mesma.

Por fim, uma classe define também uma estrutura de dados e suas possíveis operações, da mesma forma que o monitor, só que esses dados são privados a um processo ou monitor.

A linguagem CSP/K é uma extensão da linguagem de programação sequencial SP/K desenvolvida por Holt [10]. Nela foram incorporados mecanismos para permitir concorrência e exclusão mútua. A concorrência é fornecida por processos e, a exclusão mútua por monitores.

A linguagem PLZ/SYS, usada para desenvolver o sistema.

RIO, é um membro da família de linguagens para microcomputadores desenvolvida pela Zilog. É uma linguagem de programação sequencial, ou seja, não tem nenhuma construção adequada para concorrência, mas tem interface para módulos escritos em Linguagem Montadora ou em outras linguagens, além de algumas facilidades para programação em tempo real e E/S básica.

A linguagem C, desenvolvida por Kerninghan [13], foi usada para implementação do sistema Unix [21]. Essa linguagem é classificada como linguagem de alto nível orientada para máquina, já que o programador pode fazer uso direto dos recursos de "hardware", tais como registradores e endereços. Ela não fornece nenhum suporte para processos concorrentes, o sistema Unix entretanto, possui uma extensa biblioteca para ter acesso aos recursos de "hardware"; os processos são criados pelas rotinas dessa biblioteca chamadas FORK e EXEC e se comunicam por sinais ou por arquivos compartilhados.

Para sistemas multiprocessadores foi desenvolvida a linguagem Edison por Brinch Hansen [5]. Ela é baseada nas linguagens Pascal Concorrente e Módulo [28] tendo como característica principal sua simplicidade. A concorrência é fornecida por meio de um comando COBEGIN ...END, que executa os comandos em paralelo. A comunicação entre processos é implementada usando variáveis globais compartilhadas e regiões críticas, que são uma lista de comandos cercados por um comando WHEN.

O substituto das estruturas de processos, monitores e classes do Pascal Concorrente é o módulo em Edison, ou seja, ele implementa o conceito de tipos abstratos de dados.

Essas não são as únicas linguagens usadas para programação de sistemas. Appelbe [1], faz uma comparação entre as aqui colocadas e outras como Ada, Mesa, Pascal Plus, Módulo-2 e Clu descrevendo suas principais características.

CAPÍTULO III

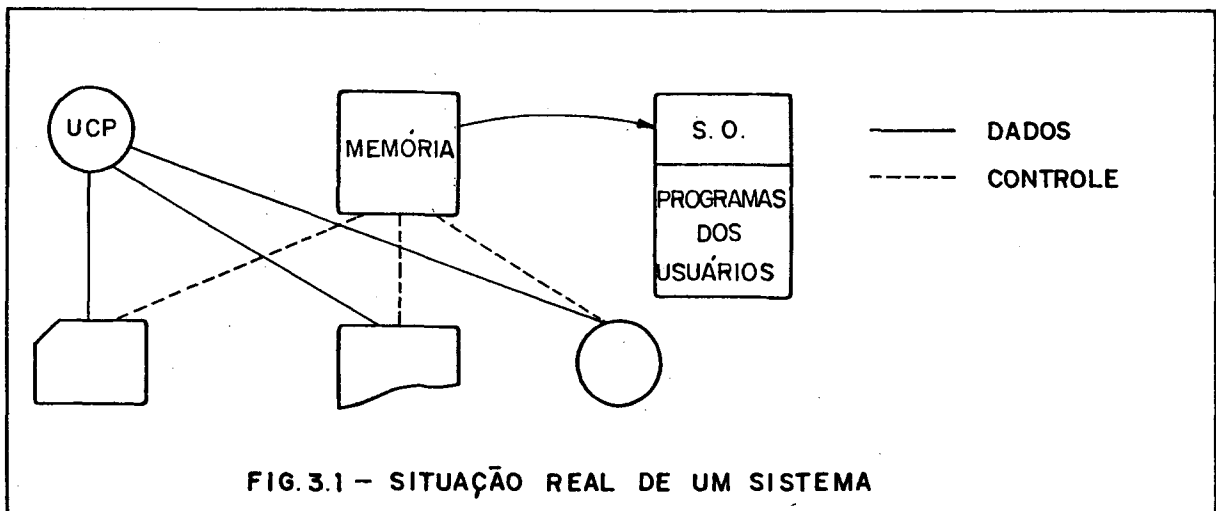
PROJETO

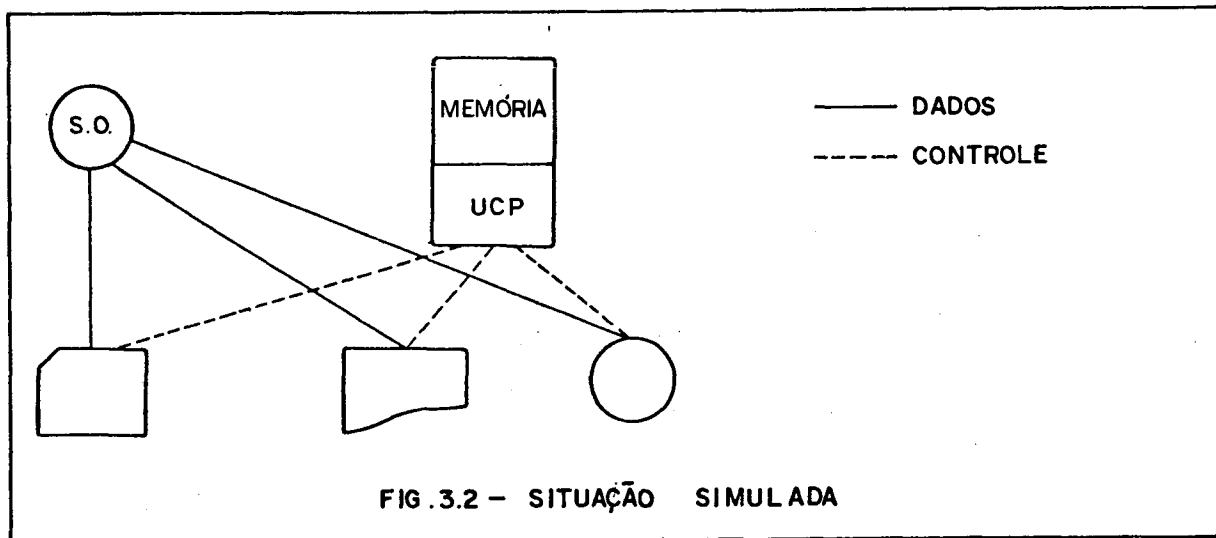
3.1 - Apresentação Global

Num sistema de computação podemos considerar dois níveis hierárquicos do ponto de vista do "software": o nível mais baixo, denominado supervisor, onde se situa o sistema operacional, e o nível mais alto, denominado usuário, onde se situam os programas dos usuários. Toda vez que um programa de usuário gera algum erro, ou condição, o processador comuta seu estado de usuário para supervisor, de forma que o erro ou condição possa ser tratado pelo sistema operacional.

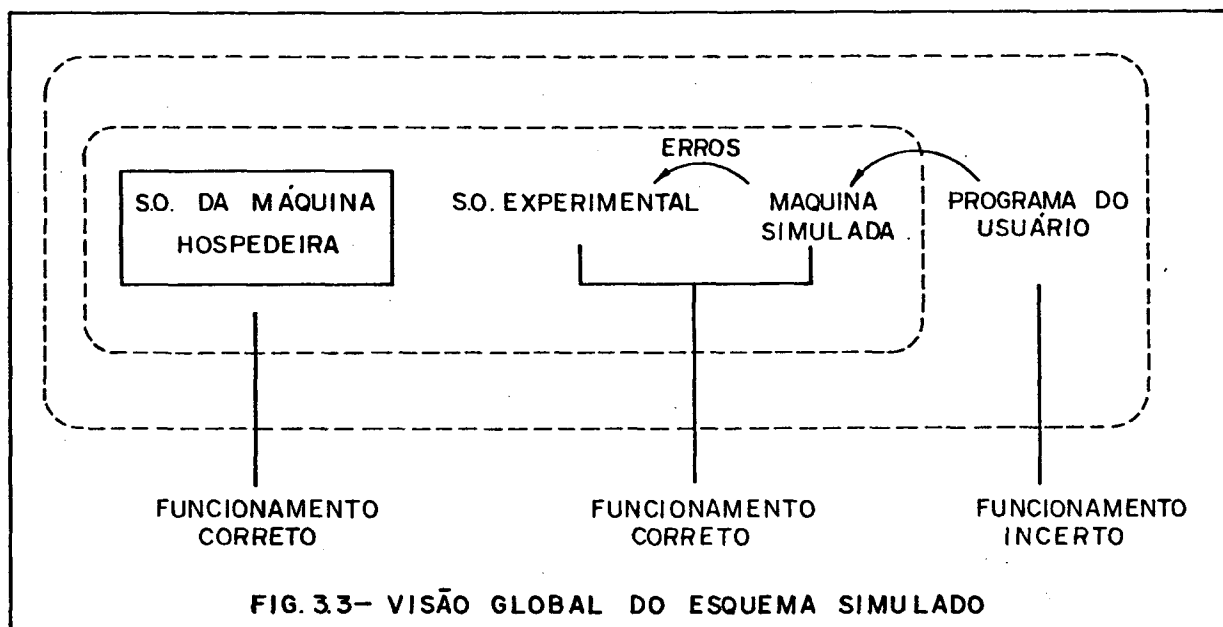
No nosso caso, onde o sistema operacional experimental funciona sobre o sistema operacional de uma máquina hospedeira e com a máquina sobre a qual os programas dos usuários são executados sendo simulada, o controle do programa do usuário é feito a cada instrução, verificando-se as possíveis condições e erros.

Os esquemas de uma situação real de um sistema de computação e da situação simulada, estão mostrados nas figuras 3.1 e 3.2 respectivamente.





O conjunto: máquina simulada, sistema operacional experimental, mais sistema operacional da máquina hospedeira, pode ser visualizado pela figura 3.3.



Nesse esquema, uma instrução do programa do usuário, só será executada, se totalmente correta, ou, se não houver nenhuma interrupção na UCP, que impeça sua execução.

Como na verdade, o sistema operacional experimental é composto pelo núcleo mais sistema operacional, uma vez detectada

qualquer condição, cabe ao núcleo, o tratamento preliminar necessário, que é a suspensão do programa com o salvamento de sua configuração e, a sinalização para o sistema operacional experimental da causa da suspensão, deixando então a ele a responsabilidade pelo tratamento complementar.

3.2 - Simulador

3.2.1 - Características Gerais

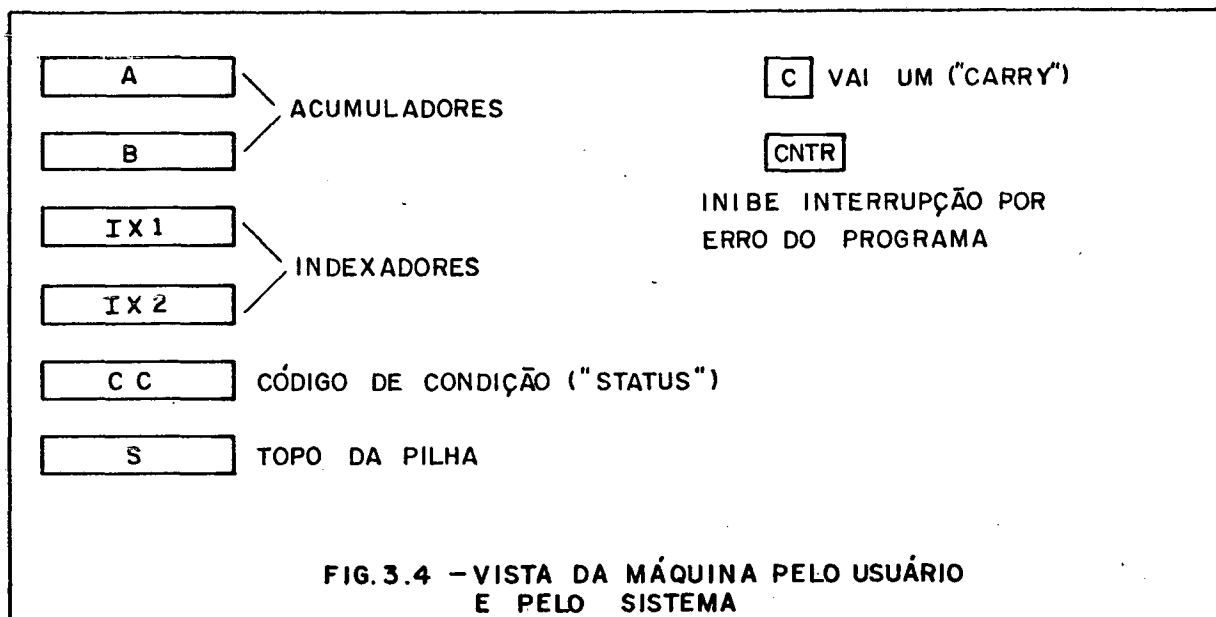
O "hardware" simulado prevê os periféricos acoplados por meio de canais, possui o suporte adequado para paginação, para o tratamento de interrupções de programa, de entrada e saída e de tempo, tendo a palavra básica composta por dezesseis bits.

Para facilitar o entendimento de sua estrutura, dividimos sua configuração em dois níveis: o nível que é visto tanto pelo usuário quanto pelo sistema e, o nível que é exclusivo do sistema.

Do ponto de vista do usuário e do sistema, a máquina é composta pelo seguinte conjunto de registradores (também mostrados na figura 3.4):

- dois acumuladores (A e B);
- dois indexadores (IX1 e IX2);
- um registrador de "status" ou código de condição (CC);
- um registrador usado para habilitar ou inibir interrupção por erro do programa (CNTR);
- um registrador de "carry" ou "vai-um" (C);
- um registrador usado para guardar o ponteiro de pilha (TPH), sendo que essa pilha é usada para armazenar endereço de retorno de subrotina.





Durante o funcionamento do simulador, as possíveis condições que podem ser detectadas são:

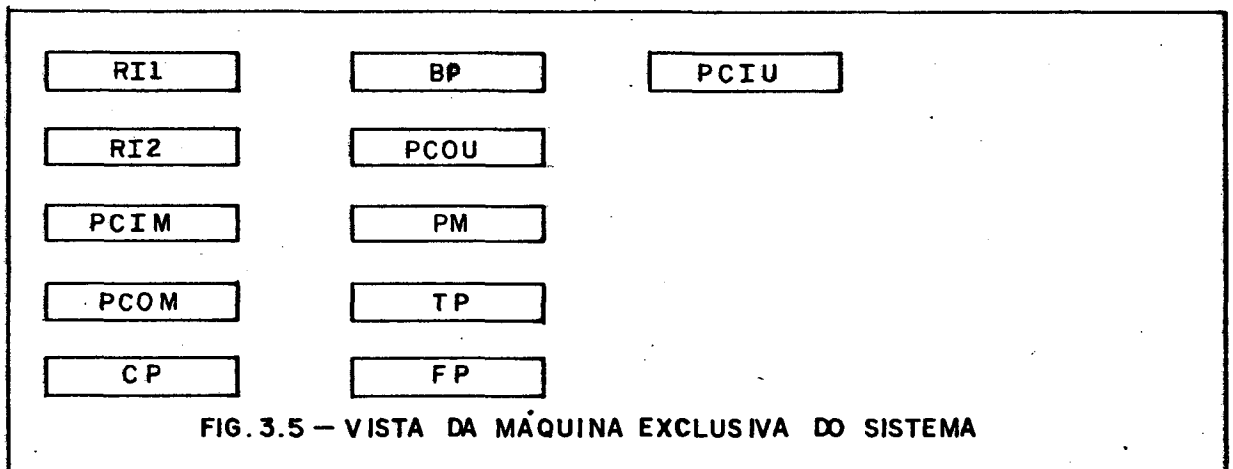
- 1 - Estouro por excesso ou "overflow";
- 2 - Estouro por falta, ou "underflow";
- 3 - Negativo, quando o resultado de uma operação gera um valor negativo;
- 4 - Zero, quando o resultado de uma operação é igual a zero;
- 5 - Vai-um ("carry"), gerado por uma instrução de deslocamento;
- 6 - Proteção de memória, gerado pela tentativa de acessar algum endereço fora dos limites permitidos para o programa em execução;
- 7 - Falta de página, gerada quando a instrução faz referência para uma página que não aquela presente na memória;
- 8 - Código de operação inválido, gerado quando a instrução solicitada pelo usuário, ou o modo de endereçamento, ou o registrador não fazem parte do repertório de instrução;
- 9 - Fim, gerado por instrução de fim de programa;
- 10- Temporizador ("timer"), gerado quando o tempo especificado para a execução do programa se esgotou;
- 11- Segmento de tempo ("timeslice"), gerado pelo relógio da UCP quando o tempo de uso contínuo da mesma se esgotou;

- 12- Interrupção de periférico, gerada por um periférico solicitando o uso da UCP;
- 13- "Overflow" na pilha, gerado por uma instrução que necessita o uso da pilha e ela está cheia;
- 14- "Underflow" na pilha, gerado pela tentativa de acesso à pilha vazia;
- 15- E/S do usuário (esusuário), gerado por uma instrução de E/S no programa.

Do ponto de vista exclusivo do sistema, temos o suporte necessário para paginação e multiprogramação. Nesse nível temos a seguinte configuração de registradores, também mostrados na figura 3.5:

- dois para instrução (RI1 e RI2);
- página corrente da instrução na memória (PCIM);
- página corrente do operando na memória (PCOM);
- página corrente da instrução do usuário (PCIU);
- página corrente do operando do usuário (PCOU);
- contador de programa (CP);
- base da pilha (BP);
- proteção de memória (PM);
- tabela de páginas (TP);
- falta de página, que guarda o número da página em falta (FP).

Os registradores PCIM e PCOU são usados para evitar-mos acesso à tabela de páginas, quando buscamos uma instrução ou um operando numa página já presente na memória. A utilização desses registradores está explicada com detalhes a seguir:



A palavra de instrução é formada pelos dezesseis bits assim divididos: oito bits para o código de operação, quatro para o modo de endereçamento e, os quatro restantes para o endereço do registrador a ser utilizado. Essa palavra de instrução é sempre mantida no registrador de instrução RI2, sendo o registrador RI1 usado como palavra de dados. Assim, uma instrução pode ser formada de uma ou duas palavras e essa informação é dada pelo código da operação. Se o último bit que forma o código de operação é zero, a instrução é de uma palavra, se um, a instrução é de duas palavras sendo necessário dois ciclos de UCP para se buscar a instrução completa.

As possíveis formas de endereçamento são: de registro, imediato, direto, indexado 1, indexado 2, relativo (ao contador de programa), e especial. Os registradores que podem ser usados por uma instrução, são os dois acumuladores e os dois indexadores.

No modo de endereçamento de registro a instrução pode ser de uma ou duas palavras. Se a instrução é de duas palavras, na palavra de dados temos, o endereço do outro registro envolvido na operação, uma vez que na palavra de instrução já temos um registro ou a palavra de dados contém o número de rotações que devem ser feitas no caso de instrução de deslocamento. A lista de todas as instruções, bem como a explicação sobre cada uma delas, encontra-se no Anexo 1.

Nos endereçamentos direto, indexado, relativo e imediato, as instruções são sempre de duas palavras. Para o endereçamento direto, o endereço do operando é o conteúdo da palavra de dados; já no indexado, na palavra de instrução, temos qual é o indexador a ser usado, e o endereço do operando é obtido, somando-se o conteúdo do indexador com o conteúdo da palavra de dados. No endereçamento relativo, o endereço do operando é dado pela soma do valor do contador de programa com a palavra de dados. Finalmente, no endereçamento imediato, na palavra de dados temos o próprio operando.

O endereçamento especial foi reservado para quatro instruções, sendo elas: instrução de nenhuma operação (NOP), que utiliza um ciclo completo de UCP, mas não afeta o programa;

instrução de fim de programa e, as instruções de E/S (leia e imprima), que não são executadas pelo simulador.

A memória está dimensionada com sessenta e quatro páginas, com trinta e duas palavras cada. Quando uma página está sendo utilizada para conter uma tabela de páginas, o primeiro bit de cada palavra, ou seja, o bit de sinal, é usado para determinar a presença ou não da página na memória. Se o bit de sinal é zero, a página se encontra na memória, caso contrário, a página não se encontra presente, e se for necessário seu uso, ela deve ser trazida do disco.

O tamanho do disco, é de mil e vinte e quatro páginas, com trinta e duas palavras cada uma.

3.2.2 - Simulação do "hardware".

O funcionamento do simulador tem início com a busca, na memória, da instrução a ser executada. Se durante a busca ocorrer alguma condição, o ciclo da máquina deve ser interrompido para iniciar o tratamento da condição. Caso contrário, vamos para a fase de execução da mesma, sendo que ao fim da execução, novamente devemos verificar a ocorrência de condições, e o mesmo procedimento acima é adotado. O simulador repete as duas etapas, busca e execução, até que a ocorrência de alguma condição, determine o seu término, para que o núcleo possa tratá-la.

O algoritmo que executa o ciclo da máquina pode ser descrito por:

Início

Repita

acabar:= falso;

BUSCA_DA_INSTRUÇÃO;

Se alguma condição for sinalizada

então acabar:= verdadeiro

senão Início

EXECUÇÃO;

Se alguma condição for sinalizada

então acabar:= verdadeiro

Fim

até acabar;
CHAMADA_DO_NÚCLEO
Fim

A fase de Busca da instrução, pode ser feita em duas etapas, já que a instrução pode ser formada por uma ou duas palavras. Além disso, temos que utilizar um mapeamento de memória para determinarmos o endereço de memória no qual, a instrução ou o dado se encontra. Esse mapeamento utiliza o contador de programa, que é um dos registradores da UCP.

O processo que vamos descrever, é válido tanto quando se está buscando a palavra de instrução, ou a palavra de dados.

Inicialmente, o contador de programa (CP), é comparado com o registrador de proteção de memória (PM). Se CP for maior que PM, o código de condição proteção de memória é sinalizado, e a busca termina. Se a condição não ocorrer, o próximo passo, é determinarmos a posição de memória em que se encontra a instrução ou o dado, conforme a fase em que estivermos. Nesse ponto, o contador de programa é usado para gerar dois valores : o número da página onde se encontra o contador de programa (PAG_CP), e o deslocamento dentro da página (DESLCP). Se PAGCP coincidir com o registrador de página corrente da instrução do usuário (PCIU), a palavra de instrução se encontra na posição de memória dada pelo registrador de página corrente da instrução na memória (PCIM), combinada com o deslocamento DESLCP, já calculado. Na situação inversa, ou seja, PAGCP não coincide com PCIU, esse número da página vai ser usado como índice para acessarmos a tabela de páginas. Se o acesso a tabela, determinar que a página procurada não se encontra na memória, o código de condição falta de página é sinalizado, o registrador de falta de página é atualizado com o número da página em falta, e a busca termina. Se a página estiver presente na memória, os registradores PCIM e PCIU são atualizados com os valores da página do contador de programa. Feito isso, o registrador de instrução/dado é carregado com o valor da posição de memória dada por PCIM combinado com DESLCP (MEM [PCIM,DESLCP]).

Ainda, nessa fase de busca, temos a execução de duas

outras rotinas. A rotina que faz a simulação de relógio, que é executada a cada acesso à memória, e a rotina que simula o tempo dos periféricos, ou em outras palavras, verifica no caso de um periférico estar ligado, se ele acabou de executar sua tarefa.

Cabe ao usuário, determinar qual será o tempo que seu programa pode ou deve demorar para ser processado. Portanto, a rotina que simula o relógio, deve, não só atualizar o relógio da UCP, como fiscalizar se o tempo dimensionado pelo usuário não se esgotou. A esse tempo, chamamos de tempo total. Mais ainda, como a UCP deve ser compartilhada entre todos os usuários, cabendo a cada um deles o uso da mesma por um tempo determinado, que chamamos aqui de tempo parcial, esse controle também é feito por essa rotina.

A atualização do tempo total determinado pelo usuário é feita decrementando-se o tempo total de um cada vez que a unidade de tempo é gasta durante o processamento.

A rotina que simula o tempo dos periféricos verifica se cada periférico está ligado e se o tempo necessário para executar sua tarefa não se esgotou. Se esgotado, ele deve gerar para o núcleo um sinal de interrupção dizendo também qual dos periféricos é o solicitante.

Os códigos de condição que podem ser gerados após a fase de busca são: tempo total esgotado, tempo parcial esgotado, interrupção de periférico, proteção de memória, falta de página, ou código de operação inválido.

Na fase de execução, como já vimos, se o modo de endereçamento é direto, indexado 1, indexado 2, ou relativo, o endereço do operando deve ser calculado. Uma vez calculado, o endereço gerado é comparado com a proteção de memória (PM), se maior, a condição correspondente deve ser sinalizada, e essa fase termina. Se a condição não for sinalizada, e a instrução for de desvio (condicional ou incondicional), ou de chamada ou retorno de subrotina, ela pode ser executada imediatamente. Se a instrução não se enquadra nas citadas, o endereço efetivo do operando deve ser calculado. O cálculo segue a mesma linha utilizada na fase de busca com o contador de programa. O endereço do

operando é usado para gerar dois valores: a página efetiva do operando (PAGEFOP), e o deslocamento dentro da página (DESLEFOP). Se PAGEFOP não coincidir com a página corrente do operando do usuário (PCOU), um acesso a tabela de páginas é necessário, para localizarmos a página em questão.

Novamente, aqui pode ocorrer a condição falta de página. Caso não ocorra, o registrador de página corrente do operando na memória (PCOM), será carregado com o valor da página obtida pelo acesso à tabela de páginas, e o registrador de página corrente do operando do usuário (PCOU), com o valor da página efetiva do operando (PAGEFOP). Após todo esse processo, a instrução pode ser executada, e o operando se encontra na posição de memória dada por MEM [PCOM, DESLEFOP].

Se o modo de endereçamento for de registro, com instrução de duas palavras, sabemos que a palavra de dados contém o endereço do segundo registrador envolvido na operação, portanto, o procedimento acima não se aplica. A situação é igual para o endereçamento imediato, já que a palavra de dados contém o próprio operando.

Após a execução de cada instrução, os testes das condições geradas pela mesma são feitos, e se necessário sinalizada a respectiva condição. No caso da instrução fazer acesso à memória, existirá também a atualização do relógio e do tempo dos periféricos.

Terminada essa fase, os códigos de condição que podem ter sido gerados são os mesmos possíveis após a fase de busca, retirando o código de operação inválido, e acrescentando, "overflow" na pilha, "underflow" na pilha, "overflow", "underflow", instrução de fim de programa, ou de Leitura/Escrita.

O procedimento execução termina, deixando o contador de programa apontando a instrução que gerou uma condição, ou, apontando a próxima instrução no caso de nada ocorrer. Na figura 3.6 a,b,c,d temos o diagrama que resume a operação do simulador, e o detalhamento de alguns dos seus blocos.

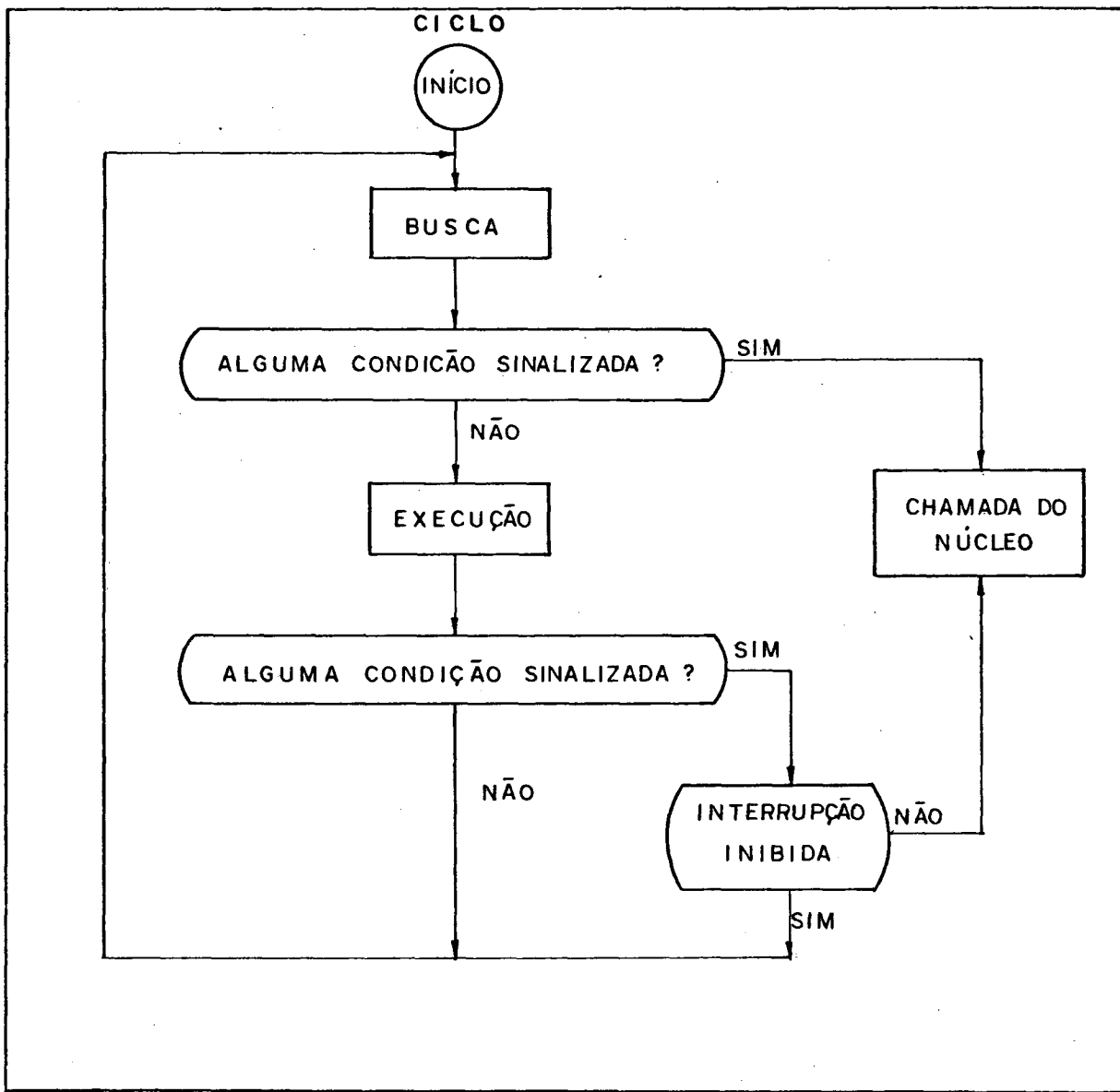


FIG.3.6a - DIAGRAMA DO SIMULADOR

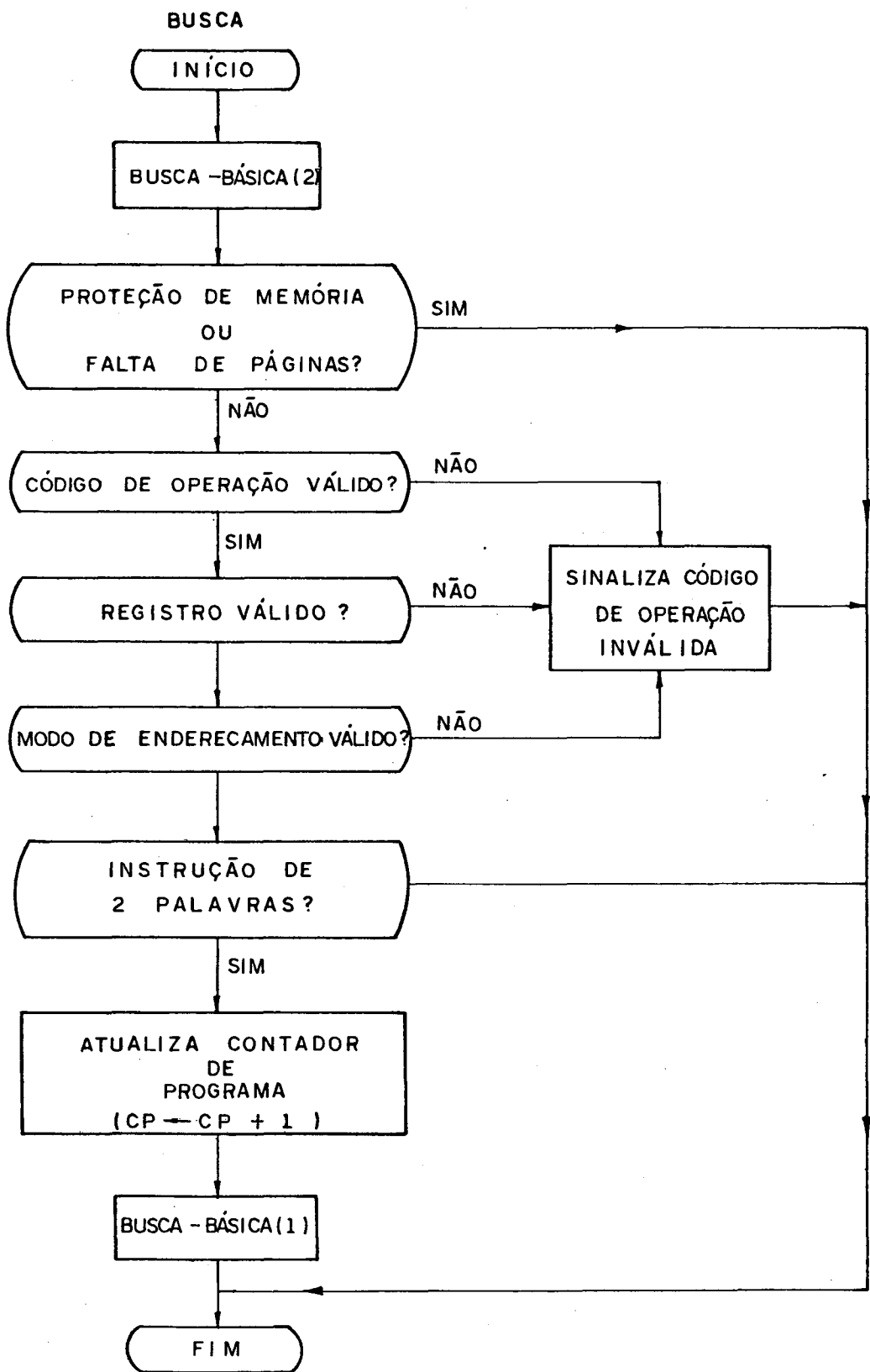


FIG.3.6b - BUSCA DA INSTRUÇÃO

BUSCA - BÁSICA (N)

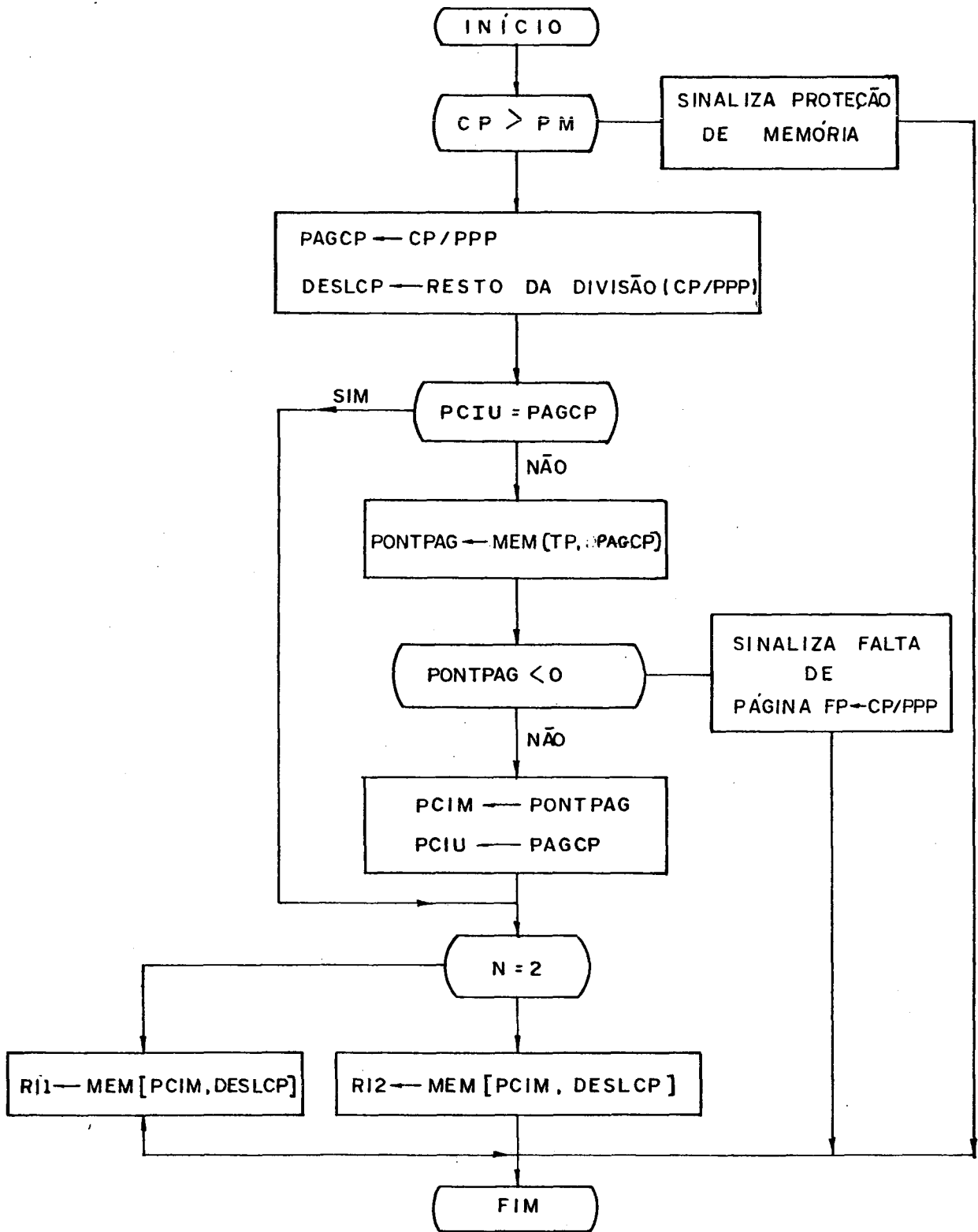


FIG.3.6c - BUSCA DA INSTRUÇÃO

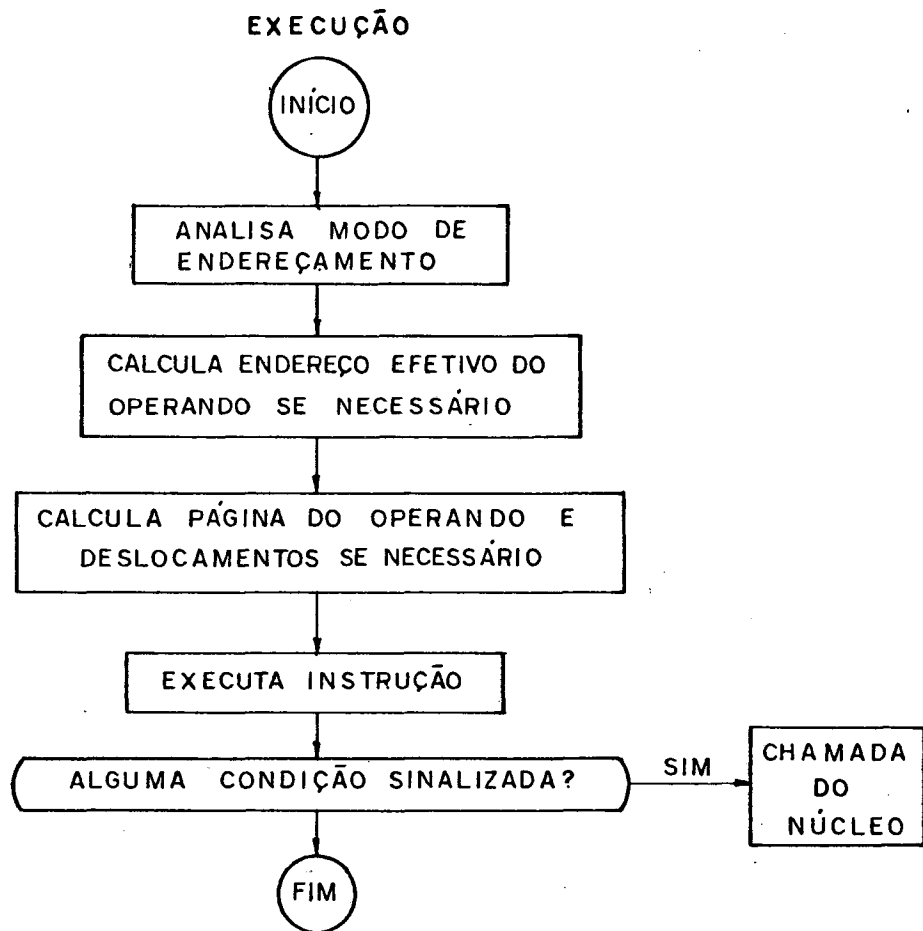


FIG.3.6d - EXECUÇÃO DA INSTRUÇÃO

3.3 - Núcleo.

Genericamente, o núcleo é o responsável pelas primitivas para sincronização e comunicação entre os processos, pelo tratamento preliminar de interrupções, pela execução de entrada/saída, escalação de curto prazo, implementação de filas de acopladores ("buffers"), manutenção de tabelas de controle, e pela criação e destruição de processos.

O núcleo manipula o programa do usuário, através de um resumo de seus elementos principais, denominado descritor de processo, que permite, tanto a colocação do programa em execução, quanto sua retirada. No nosso caso, o descritor de processo, contém:

- 1 - Endereço da Tabela de páginas;
- 2 - Conteúdo do acumulador A;
- 3 - Conteúdo do acumulador B;
- 4 - Conteúdo do indexador IX1;
- 5 - Conteúdo do indexador IX2;
- 6 - Conteúdo do contador de programa CP;
- 7 - Endereço da página faltosa;
- 8 - Endereço do registrador usado para E/S;
- 9 - Código de condição;
- 10 - Conteúdo do registrador C;
- 11 - Conteúdo do registrador CNTR;
- 12 - Endereço do topo da pilha, ou seja, conteúdo do registrador TPH;
- 13 - Conteúdo das posições da pilha.

Para que um programa seja executado, ou seja, ganhe o controle da UCP, o núcleo carrega os registradores da UCP com o descritor do processo. Após isso, o simulador é chamado, passando dessa forma o controle para o programa, e então ele é executado.

A nível do núcleo temos: uma fila que guarda os processos do sistema operacional que estão prontos para serem executados; uma "fila" de uma posição, que contém o descritor do processo do usuário que está pronto para ser executado, onde o processo do sistema operacional chamado agendador tem acesso ;

e três posições que guardam o endereço dos processos que estão suspensos esperando o término da execução de uma operação de E/S, sendo que, cada uma dessas posições está associada a um periférico.

O núcleo só aloca a UCP para um processo do usuário, quando não existir nenhum processo do sistema pronto para rodar. No caso de não haver nem processo do sistema e nem do usuário, a UCP fica ociosa, esperando por interrupção de algum periférico.

O sistema operacional se comunica com o núcleo, através de dois procedimentos que são: atrasa na fila, e continue processo que espera na fila. O procedimento "atrasa", é sempre executado, quando o processo entra num monitor, e necessita de uma condição para prosseguir, que no momento não é satisfeita, dessa forma, o processo será colocado na fila associada a condição, e outro processo será escalado para rodar. O procedimento "continue" é sempre o último comando executável dentro do monitor, ele sinaliza a fila associada a uma condição, que ela foi satisfeita. Dessa forma, se a fila não estiver vazia, um dos processos que aguardavam pela condição passa a ser executado.

Os algoritmos para os procedimentos atrasa e continue são dados por:

PROCEDIMENTO ATRASA(C);

Início

Pega o processo corrente e coloca na fila associada à condição C;

Se a fila dos processos do sistema estiver vazia

então Se não existir processo do usuário escalado

então UCP entra em estado de espera

senão Coloca processo do usuário para rodar

senão pega um processo da fila dos prontos do sistema e coloca para rodar

Fim

PROCEDIMENTO CONTINUE (C);

Início

Se a fila C estiver vazia

então Se a fila dos processos do sistema estiver vazia

então o processo que sinalizou continua sua execução

senão escala um processo do sistema para rodar

senão Início

coloca o processo corrente na fila dos prontos do sistema;

escala um processo da fila C para rodar

Fim

Fim;

As possíveis causas que provocam uma interrupção na UCP, quando ela está rodando um processo do usuário, foram divididas em três classes: na primeira estão interrupção do periférico, proteção de memória e código de operação inválido; na segunda estão, tempo total esgotado, falta de página, E/S do usuário, "overflow", "underflow", instrução de fim de programa, "overflow", na pilha, "underflow" na pilha; e na terceira, tempo parcial (segmento de tempo) esgotado.

A interrupção de periférico também pode ocorrer com a UCP em estado de espera.

O núcleo atende primeiro, as interrupções de primeira classe, e entre elas, a interrupção de periférico é a de maior prioridade.

Nessa situação, o núcleo suspende a execução do programa rodando (se existir), vai na posição correspondente ao periférico que interrompeu, e coloca o endereço contido na posição, na fila dos processos do sistema que estão prontos. O programa que estava rodando, fica na posição do programa escalado para rodar, já que a causa de sua suspensão, não foi gerada por ele, dessa forma, quando a UCP puder ser alocada a um usuário, ele novamente ganha seu uso.

Pela estrutura do simulador, pode ocorrer simultaneamente com a interrupção do periférico, proteção de memória e tempo total esgotado, ou, proteção de memória e tempo parcial esgotado. Como a condição proteção de memória tem prioridade

sobre as duas interrupções de tempo, com relação a seu tratamento, a simultaneidade de ocorrência não interfere no tratamento que deve ser dado ao programa. A crítica que podemos fazer quando ocorre essa simultaneidade, é que quando o programa retornar a UCP ele novamente vai ser retirado pela condição proteção de memória, que já poderia ter sido detectada e tomada a atitude pertinente. Essa é uma melhoria que poderá ser introduzida no simulador. A mesma situação pode ocorrer com a condição de operação inválida

Para as interrupções de segunda classe, a prioridade é dada a condição tempo total esgotado. Detectada essa condição são canceladas as interrupções da mesma classe que podem ocorrer simultaneamente, que são "overflow", "underflow", e E/S do usuário. O descritor de processo é assinalado com a causa da suspensão, e sinalizado o processo do sistema operacional que faz o tratamento de interrupção.

As ações tomadas pelo núcleo no tratamento das outras interrupções da segunda classe, são iguais para todas. Cancela a interrupção de terceira classe, assinala no descritor do processo a causa da interrupção, e sinaliza o processo do sistema operacional que faz o tratamento da interrupção.

Para a interrupção da terceira classe, o procedimento é o mesmo feito no final de cada classe, ou seja, assinala o descritor e sinaliza.

Com exceção da interrupção do periférico, qualquer outra condição faz com que o programa que estava sendo executado deixe o núcleo, devendo portando ser escalado outro processo de usuário.

Como já descrevemos anteriormente, temos três periféricos: leitora de cartões, impressora, e disco, cada um ligado ao seu respectivo canal. Os comandos de entrada/saída, que iniciam a transferência de dados, são executados através da subrotina que chamamos de ES. Essa subrotina, ativa o respectivo canal, e inicia a transferência de dados. A partir daí, o canal e a UCP prosseguem independentemente um do outro. A UCP está livre para outro processamento, e o canal continua a monitorar a transferência de dados. Quando a transferência termina, o canal sinaliza uma interrupção.

Antes da chamada da subrotina ES, o processo primeiro deve ganhar o acesso ao periférico, ou seja, a rotina só é chamada, se o periférico estiver livre. O processo que quer se utilizar de um periférico deve portanto primeiro solicitar o uso do mesmo. Se o periférico não estiver livre, o processo é suspenso numa fila onde ficam os processos que querem utilizá-lo, temos dessa forma, uma fila associada a cada periférico. Após o término do tratamento da interrupção, o periférico é liberado, sinalizando a respectiva fila que outro processo já pode utilizá-lo.

Uma instrução de E/S de usuário gera uma interrupção, que faz com que um processo do sistema operacional, que chamamos de ESUSUARIO seja executado. O núcleo, nesse caso, somente é o responsável pela suspensão do programa, passando para o sistema, o tipo da instrução, ou seja, se o usuário deseja ler ou escrever.

Quando um processo é suspenso na fila associada ao periférico, o núcleo se encarrega de escalar um novo processo para rodar.

3.4 - Sistema Operacional

O sistema operacional é formado por um conjunto de processos concorrentes, que se comunicam através de monitores. O programa do usuário se configura para o sistema como um Bloco de Controle de Programa (BCP), que além das informações contidas no Bloco Descritor (que é usado pelo núcleo), contém toda a configuração do programa a nível do disco. As informações complementares que estão no BCP são:

- Identificador do usuário;
- Proteção de memória (última posição de memória cujo acesso é permitido para o particular usuário);
- Endereço inicial do programa no disco (endereço da página);
- Tamanho do programa em páginas;
- Endereço inicial da área de dados;
- Apontador corrente da área de dados;

- Endereço final da área de dados;
- Endereço inicial da área de impressão;
- Apontador corrente da área de impressão;
- Endereço final da área de impressão;
- Sinalizador da condição de término do programa;
- Tipo de operação de entrada e saída a ser realizada.

O usuário é quem fornece ao sistema, além do seu identificador, o tempo máximo que o seu programa deve demorar, e o número de páginas que serão utilizadas para os resultados (área de impressão).

O programa do usuário deve ser fornecido da seguinte maneira:

```

JOB                ( Identificador de controle de início de ta-
                   refa)
identificador do usuário
número de páginas para impressão
PGR                ( Identificador de controle de início de pro-
                   grama)
:
:
instruções do programa
:
:
DAD                ( Identificador de controle de início de da-
                   dos)
:
:
dados do programa
:
:
EOJ                ( Identificador de controle de fim de tarefa)

```

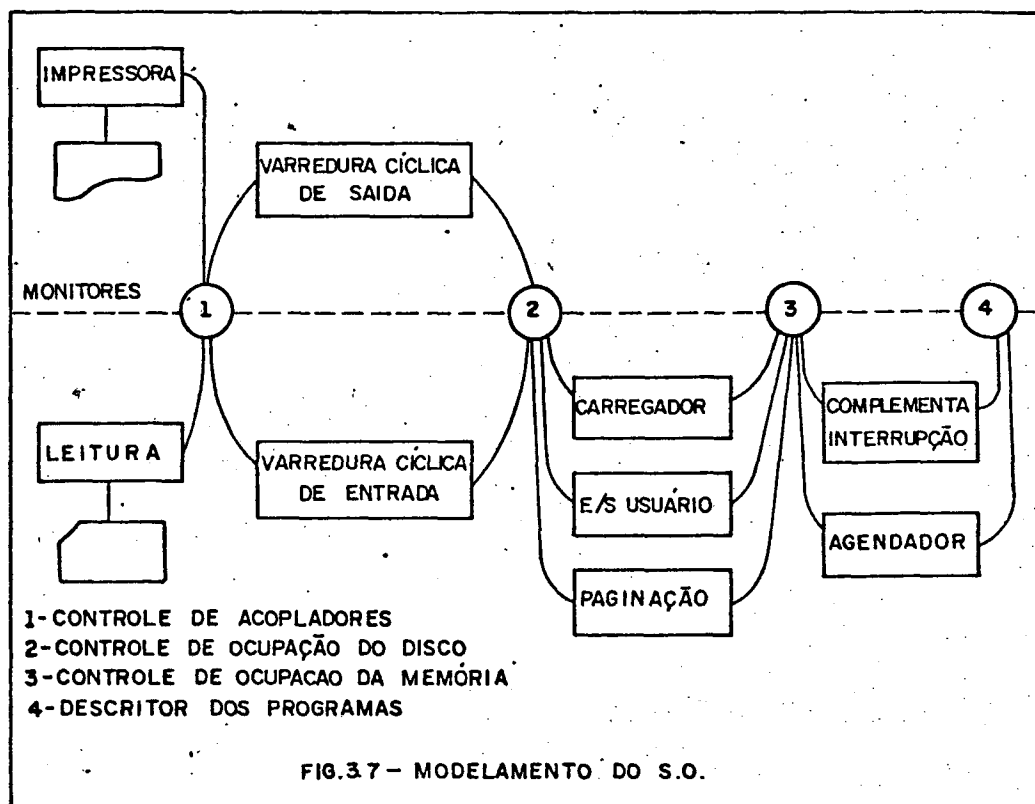
Os processos do sistema operacional podem ser divididos em função das fases pelas quais um processo do usuário deve passar, e que são as relacionadas a seguir:

a) Fase de entrada

- a.1) leitura: preenche um acoplador ("buffer")
- a.2) varredura cíclica ("spool") de entrada: analisa acoplador preenchido pela leitura, e o transfere para o disco, nesse processo, temos a criação do BCP.

- a.3) carregador: monta a tabela de páginas, e a carrega juntamente com a primeira página na memória.
- b) Fase de execução
- b.1) paginação: busca no disco a página que falta e a carrega na memória.
- b.2) E/S do usuário (ESUSUÁRIO): faz a leitura de dados na área de dados do disco, ou a impressão de resultados de impressão.
- b.3) complementa interrupção: recebe o descritor do processo que rodava quando ocorreu a interrupção e, atualiza o BCP.
- b.4) agendador: pega um BCP, transforma-o em descritor e envia ao núcleo (programa escalado para rodar).
- c) Fase de saída
- c.1) varredura cíclica ("spool") de saída: preenche um acoplador ("buffer") com uma página do disco.
- c.2) impressão: imprime a página liberando o acoplador("buffer").

O modelamento do sistema é melhor visualizado pela figura 3.7 a seguir.



Nessa figura temos assinalados quatro monitores, sendo que o monitor marcado com o número 1, faz o controle dos acopladores ("buffers") disponíveis, o monitor 2 faz o controle de ocupação do disco, o monitor 3 faz o controle de ocupação da memória, e o monitor 4 faz a sincronização entre o processo agendador e o processo complementa interrupção.

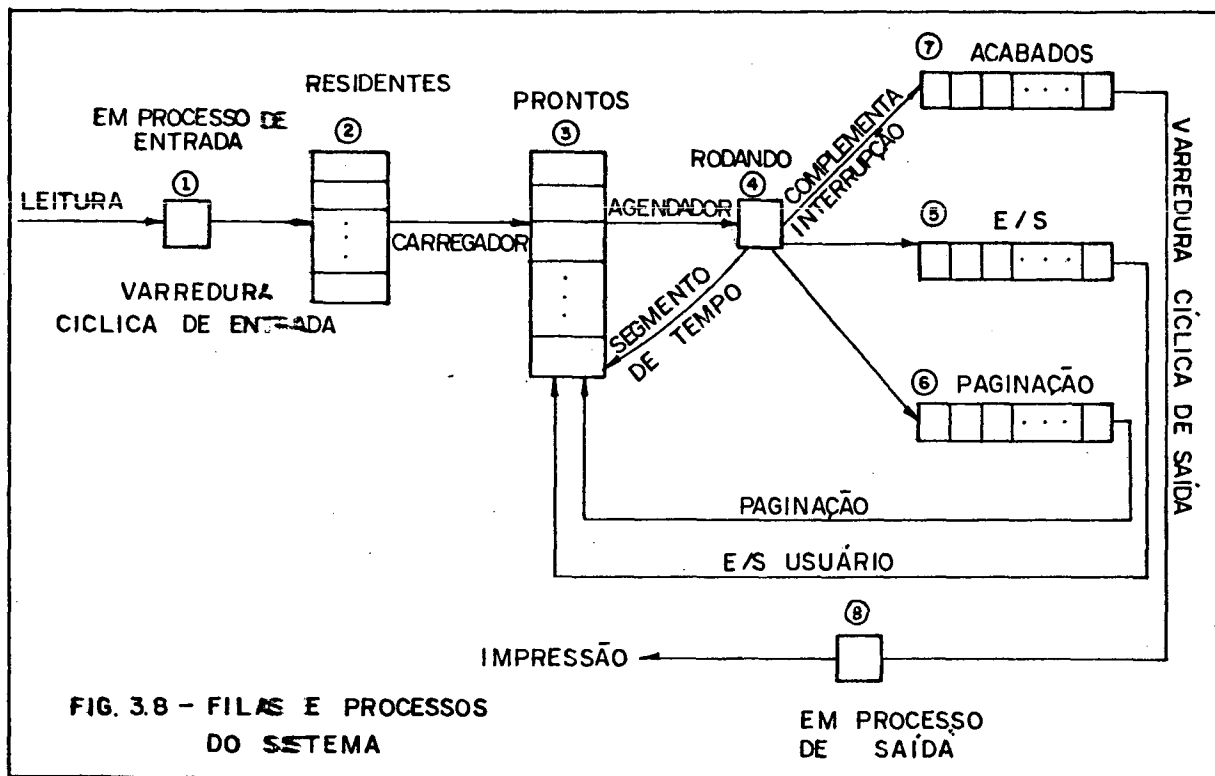
Para o controle dos acopladores ("buffers") são mantidas três filas: a fila contendo todos os acopladores ("buffers") disponíveis, a fila que contém os acopladores ("buffers") em uso pelo processo varredura cíclica ("spool") de entrada, e a fila dos acopladores ("buffers") preenchidos pela varredura cíclica ("spool") de saída.

No início, quando o sistema começa a funcionar, todos os acopladores estão disponíveis e portanto na fila dos livres (temos dezesseis acopladores). Os processos que necessitam de acopladores são: o de leitura, que preenche o acoplador e passa-o para a varredura cíclica de entrada, e a varredura cíclica de saída que preenche-o com página do disco e passa-o para o processo de impressão. O processo de impressão e a varredura cíclica de entrada liberam os acopladores colocando-os novamente na fila dos disponíveis depois que acabam de usá-los. O tamanho dos acopladores é igual ao das páginas de memória e disco.

O controle de ocupação de memória e de disco, são feitos através de filas que contém as páginas disponíveis (uma fila para o disco e outra fila para a memória).

O sistema mantém filas dos BCP's para os possíveis estados que um programa pode estar. A figura 3.8 a seguir mostra, além dessas filas, quais os processos do sistema que atuam em cada uma delas.





Além das filas indicadas no diagrama, temos a fila onde ficam os BCP's livres.

As filas 1, 4, e 8, só tem uma posição, já que só pode haver um programa entrando, um programa sendo rodado e, um programa em processo de saída. Nas outras filas podemos ter vários BCP's.

Nas setas que indicam a passagem do programa rodando para as filas, acabados, suspensos por E/S, paginação e prontos, é o processo "complementa interrupção" quem atua.

No início do sistema, todos os processos tentam funcionar, mas somente o processo de leitura vai poder atuar, isso porque a varredura cíclica de entrada vai tentar pegar um acoplador da fila dos preenchidos pela leitura, e ela está vazia, e o processo será suspenso até que um acoplador seja colocado na fila. O carregador vai tentar pegar um BCP da fila dos residentes e será suspenso porque a fila está vazia. O processo de paginação, E/S usuário, varredura cíclica de saída e impressão também serão suspensos, já que as filas em que cada um atuarão vazias. Dessa forma, somente a leitura vai poder iniciar seu trabalho.

O processo de leitura pede um acoplador para a fila dos acopladores livres. Se a fila tiver mais do que um acoplador

disponível, o processo ganhará o recurso. A explicação para essa restrição é simples. Suponhamos que seja alocado para o processo de leitura, o último acoplador disponível. Se o disco já estiver cheio, ou seja, não existe página disponível, a varredura cíclica de entrada não pode ser executada. O único processo que libera o espaço em disco é o de varredura cíclica de saída, que necessita de acoplador para poder copiar a página a ser liberada. Se não tivermos acoplador disponível, ele não poderá ser executado e teremos configurada a situação de bloqueio perpétuo ("deadlock"), pois a varredura cíclica de saída não consegue esvaziar o disco por falta de acoplador, e a varredura cíclica de entrada não consegue liberar o acoplador por falta de espaço em disco.

Uma vez que o processo ganhou o acoplador, haverá a solicitação de uma operação de E/S para que o mesmo seja preenchido. O processo então será suspenso até que a leitora complete sua tarefa. Ao término do trabalho da leitora, a UCP será interrompida, o processo de leitura ativado novamente, e ele colocará o acoplador na fila dos acopladores preenchidos pela leitora, ativando dessa forma a varredura cíclica de entrada.

Quando a varredura cíclica de entrada começa sua execução e acha o identificador de controle de início de programa, ele solicita uma BCP da fila de BCP's livres para criar o processo a nível do sistema operacional. No caso de não haver BCP livre, o processo será suspenso até que alguma seja liberada, caso contrário, o processo continua. A varredura cíclica de entrada, encerra a fase de entrada de um programa ao ler o identificador de controle de fim daquele programa. Após isso, libera o acoplador e passa o BCP para a fila dos processos residentes, e conseqüentemente sinaliza o carregador que ele já pode trabalhar.

O carregador pede uma página de memória para poder montar a tabela de páginas do programa. Se não houver página de memória livre, o processo é suspenso até que uma página seja liberada, caso contrário, começa a montá-la. Em seguida, solicita outra página livre, para carregar a primeira página do programa podendo ocorrer a mesma situação quando do pedido para a tabela de páginas. O carregador termina, mandando o BCP para a fila

dos prontos (do usuário), e permite então que o agendador comece seu trabalho.

O agendador pega o BCP, transforma-o em descritor, isto é, prepara a criação do processo ao nível do núcleo, e envia-o para o mesmo. Quando não houver processos do sistema para serem processados, o processo do usuário que o agendador preparou, ganha o uso da UCP. Em execução pela UCP, qualquer condição ou interrupção faz com que o núcleo suspenda a execução do mesmo, salvando o seu estado no descritor, e sinalizando o processo "complementa interrupção" para que ele faça o tratamento necessário.

O processo complementa interrupção atualiza o BCP com o descritor fornecido pelo núcleo, analisa a causa da suspensão, passando o BCP para a fila correspondente a causa. Se o processo foi suspenso por tempo parcial esgotado, seu BCP volta para a fila dos prontos, e se o escalador estava esperando por BCP nessa fila, ele será disparado. Se a suspensão foi causada por uma instrução de E/S, o BCP será colocado na fila 5 (suspensos por E/S), podendo então ser iniciado o processo que faz a entrada ou saída do usuário.

O processo E/S usuário, é o responsável pela execução da instrução pedida, e termina colocando o BCP de volta na fila dos prontos.

No caso do processo do usuário ter sido suspenso por falta de página, o processo "complementa interrupção", coloca-o na fila 6 e o processo de paginação pode ser ativado. A paginação, faz o tratamento da falta de página, carregando a página faltosa na memória e transfere o BCP para a fila dos prontos.

O processo complementa interrupção coloca o BCP na fila dos acabados, se a condição que gerou a suspensão do programa foi um erro ou simplesmente uma instrução de fim. Nesses casos, antes de mover o BCP para a fila de acabados, as páginas de memória que o usuário estava ocupando são liberadas já que ele não mais retornará para execução. Após mover o BCP para a fila de acabados, o processo de varredura cíclica de saída pode funcionar, e é ele quem libera as páginas do disco que foram alocadas para o usuário, e coloca o processo de impressão em condições de iniciar sua tarefa.

Os processos do sistema são cíclicos, ou seja, se repetem indefinidamente. Quando um processo move um BCP para uma fila, o monitor dá um sinal de continue para tal fila, de forma que se tiver um processo esperando por essa condição, ele vai ser disparado e o processo que sinalizou vai para a fila dos prontos do sistema, estando novamente em condições de ser escalado.

A leitura preenche um acoplador com uma instrução em cada posição. As instruções do programa do usuário, bem como as instruções de controle do programa, são escritas em mnemônicos e são transformadas em códigos de máquina pelo próprio processo de leitura. É lógico que esse não é o procedimento feito numa máquina real, mas aqui foi adotado para simplificar o processo.

3.5 - Aspectos da Implementação

Como foi colocado na introdução, a falta do compilador da linguagem Pascal Concorrente, fez com que implementássemos esse sistema em Pascal. Como nessa linguagem não temos tipos abstratos de dados, todas as variáveis usadas tanto no sistema quanto no núcleo, são globais e portanto devem ser manipuladas com cuidado. Além disso, existe o problema de comunicação entre o núcleo e o sistema, já que o núcleo é o responsável pela escalação, e também pela suspensão dos processos do sistema.

Para explicar a implementação, vamos usar um dos processos do sistema como exemplo. Vamos analisar detalhadamente um trecho do processo carregador. Ele inicia sua execução, pedindo BCP da fila 2. Se a fila estiver vazia o processo é atrasado numa fila, que deve ser sinalizada assim que algum BCP for colocado na fila 2. Quando receber esse sinal, que nada mais é que um continue na fila, o escalador deve retornar para execução e o processo que sinalizou colocado na fila de prontos do sistema.

De posse de um BCP, o carregador vai solicitar página de memória livre, para montar a tabela de páginas. Nesse ponto o processo pode ser suspenso se não existir página de memória livre. Quando o processo puder ser executado novamente, ele

retoma a execução, a partir do ponto de suspensão.

Prosseguindo a execução, novamente é solicitada uma página de memória, agora para carregar a primeira página do programa do usuário. De novo pode ocorrer nova suspensão do processo. Conseguida a página, deve ser copiado o conteúdo da página do disco para ela. Para isso, primeiro devemos fazer a solicitação do uso do disco, e se ele estiver ocupado, acontece nova suspensão nesse ponto. Caso contrário, ganha o uso do recurso, e é suspenso até que o disco tenha completado sua tarefa.

O processo continua até que ele move o BCP para a fila 3, deixando então o escalador em condições de funcionar.

O que realmente nos interessa nesse exemplo é analisar, como o núcleo deve ser informado na hora de ativar um processo, a partir de que ponto ele deve ser retomado.

A solução adotada foi fazer a chamada de procedimento de monitores ser precedida por um rótulo, sendo que esse rótulo substitui a configuração da UCP. Se a condição para o processo prosseguir sua execução não for satisfeita, esse rótulo será colocado na fila associada a condição, e será escalado outro rótulo da fila dos prontos. Logicamente, esse rótulo é passado como parâmetro na chamada do procedimento, e a suspensão do processo e ativação de um novo já foi visto na descrição do procedimento atrasa. Quando um rótulo é escalado de uma fila, é feito um desvio para o ponto do programa aonde ele se encontra. Dessa forma, conseguimos implementar a concorrência entre os processos que compõem o sistema.

CAPÍTULO IV

O mecanismo de agendamento de processos num sistema multiprogramado é um dos pontos fundamentais, já que é ele quem é o responsável pela distribuição da UCP entre os vários processos que a utilizam. A UCP, além de rodar processos do usuário e do sistema, deve responder a requisições dos processos e, interrupções dos periféricos.

Para um sistema com uma única UCP como é o nosso caso, nunca podemos ter mais do que um processo em execução. Se existirem vários processos para serem executados, eles devem esperar pela UCP, até que ela fique livre. Os processos que estão a espera ficam colocados em uma fila que chamamos de fila de processos prontos para serem executados.

No nosso sistema como já visto, temos duas filas de processos prontos, a fila que contém os processos do sistema e a fila de processos dos usuários. Entre essas filas, existe, um esquema implícito de prioridade, já que um processo do usuário, só ganha o uso da UCP, quando não existir nenhum processo do sistema em condições de utilizá-la, ou seja, a fila dos processos do sistema deve estar vazia.

Quando um processo do sistema, ganha o uso da UCP, ele permanece em execução, até que a necessidade de um recurso não disponível faça com que ele deixe-a, ficando na fila associada ao recurso, esperando sua vez de utilizá-lo. Em outras palavras, os processos do sistema não são interrompíveis, e não temos nenhuma restrição com relação ao tempo de utilização da UCP pelos mesmos.

Para os processos do usuário, existe definido um intervalo de tempo que o processo pode permanecer utilizando a UCP. Esse intervalo de tempo é normalmente chamado de segmento do tempo ("timeslice") e aqui estamos chamando de tempo parcial. Quando o processo vai começar sua execução, o segmento de tempo é iniciado, e é gerada uma interrupção ao fim do mesmo. Essa interrupção, como sabemos, não é a única que pode ocorrer e

qualquer outra causa, faz com que o processo deixe a UCP sem ter utilizado todo o tempo parcial a que tinha direito.

Toda vez que um processo deixa a UCP, com exceção da interrupção de periférico, o agendador trabalha, deixando um novo processo em condições de utilizá-la. No caso da interrupção do periférico, o processo deixa a UCP, e no seu descritor também é guardado o segmento de tempo que restava. Sendo ele o próximo a utilizar a UCP, com esse procedimento ele a usará somente o tempo que lhe restava.

No sistema estamos manipulando a fila dos processos prontos como uma FILA FIFO ("first in, first out"), ou seja, os novos processos são colocados no fim da mesma, e o agendador sempre retira o primeiro elemento (Algoritmo "Round - Robin").

O desempenho desse tipo de agendamento, está fortemente relacionada com o segmento de tempo definido. Se for muito grande, o tempo de resposta dos processos pode variar de acordo com a ordem de entrada dos mesmos. Como exemplo, suponha três processos definidos e seus respectivos tempos de execução dados na tabela abaixo:

PROCESSOS	TEMPO DE EXECUÇÃO
1	24
2	3
3	3

Se o segmento de tempo for igual a 24, a ordem de entrada dos processos for 1, 2 e 3, e não ocorrer nenhum erro ou condição durante a execução dos mesmos, o tempo de resposta para os processos seriam 24, 27, e 30 respectivamente.

Caso a ordem de entrada seja alterada para 2, 3 e 1, teríamos como tempo de resposta 3, 6 e 30 respectivamente.

Sabemos que quando a UCP está rodando o programa de um usuário, e recebe alguma interrupção, o núcleo deve suspender a execução do programa. A suspensão, implica no salvamento de seu descritor, da mesma forma que para a execução, os registradores da UCP são carregados com informações contidas no

descriptor. Se o tempo parcial definido for muito pequeno, a UCP pode ser sobrecarregada com a tarefa de salvamento e colocação de novo processo. O tempo gasto nesse procedimento, varia de máquina para máquina, e depende da velocidade de acesso à memória, do número de registradores envolvidos e do tipo de instruções utilizadas para esse fim. Dessa forma, o segmento de tempo deve ser dimensionado de tal forma, que o sistema não degenera para nenhum dos dois extremos. Esse tempo é geralmente da ordem de 10 a 100 milissegundos [Peterson, 19].

Dentro da forma como foi tratado o problema de agendamento, alguns melhoramentos podem ser feitos com relação ao tratamento da interrupção do periférico. Como o procedimento adotado, simplesmente salva o descriptor e o segmento de tempo que resta sem nenhuma restrição, corremos o risco desse tempo estar praticamente esgotado, ou em outras palavras, quando o programa voltar para execução, imediatamente gerará uma interrupção por segmento de tempo esgotado. A solução para esse caso é o teste do tempo restante, se for muito pequeno, adota-se o procedimento normal que inclui a execução do agendador.

Todas as filas existentes no sistema são tratadas como FIFO.

O agendador pode ser sofisticado, permitindo que os processos do usuário tenham prioridades de execução entre eles. Por exemplo suponha que temos categorias distintas de usuários, cada uma delas com uma prioridade, a UCP será sempre alocada ao processo com maior prioridade, e, como podemos ter processos com prioridades iguais, o esquema FIFO pode ser usado para esses processos.

A formação da fila dos processos prontos pode ser feita de duas maneiras. A primeira é mantê-la ordenada de acordo com a prioridade, e dessa forma existe o trabalho adicional envolvido na inserção de novos processos. A outra forma é colocar os novos processos sempre no fim da fila, e deixar ao agendador a tarefa de percorrê-la e selecionar o processo com maior prioridade.

Os processos com menor prioridade só serão executados quando não houver nenhum processo de outra categoria pronto

para execução. Com isso, podemos perceber que o risco de termos processos que nunca ganharão o uso da UCP é bastante alto. Uma solução para contornarmos esse problema, é fazermos um incremento em intervalos de tempo na prioridade dos processos que permanecem na fila. Assim, consegue-se garantir que em algum instante, mesmo os processos com baixa prioridade ganharão o uso da UCP.

Existem várias outras políticas de escalção que podem ser utilizadas e são amplamente discutidas na literatura.

No capítulo 3, foi discutido a forma como é localizada uma instrução ou um operando na memória. Vimos também, que se a página do contador de programa não corresponde ao registrador de página corrente da instrução do usuário, ou se a página do operando não corresponde ao registrador de página corrente do usuário, deve-se fazer um acesso à tabela de páginas, e esse acesso pode determinar que a página que está sendo endereçada não se encontra presente na memória. Essa situação leva a condição falta de página que indica ao sistema que a página faltante deve ser trazida do disco e carregada na memória. O número da página faltante é passado para o sistema pelo núcleo no descritor do processo.

O processo de paginação é o responsável por localizar no disco a página e trazê-la para a memória. A localização é feita utilizando-se o endereço que foi passado pelo núcleo. Uma vez localizada, se existir página de memória não usada, o processo de paginação se resume a simplesmente copiá-la, atualizar a tabela de páginas indicando agora sua presença na memória e passar o processo para a fila dos prontos.

No caso de não haver página disponível na memória a solução é substituir alguma página da memória pela página faltante. Toda vez que é processada uma substituição, a página que vai ser retirada é recopiada no disco, já que ela pode ter sido alterada e a página faltante então copiada em seu lugar.

O problema principal do processo de paginação é como escolher a página que deve ser substituída. Para resolver essa questão já foram desenvolvidos vários métodos.

O método mais simples é conhecido como FIFO ("First

In, First Out"), e como o próprio significado do nome diz, a página mais antiga na memória é a candidata a substituição. Ele é facilmente implementado através de uma fila, onde as novas páginas alocadas são sempre adicionadas no fim da fila, sendo desse modo escolhida para substituição, a página que está no início da mesma.

O desempenho do sistema que utiliza essa política de substituição de páginas nem sempre é boa. A página que foi carregada primeiro, pode ser uma página que esteja sendo utilizada com muita frequência, o que pode provocar rapidamente uma nova paginação. Existe ainda a possibilidade da primeira página da fila ser a própria página corrente, o que também fatalmente levará a uma nova paginação, atrasando então a execução do programa.

Esse algoritmo foi implementado no nosso sistema, com a ressalva de que as páginas alocadas para montar as tabelas de páginas não participam da fila, já que elas não podem ser substituídas.

Para contornar os problemas gerados pelo algoritmo FIFO, o algoritmo conhecido por LRU ("Least Recently Used") pode ser usado. Aqui, a idéia é substituir a página que está a mais tempo sem ser utilizada. Nesse caso, temos que associar a cada página o tempo da última vez que ela foi utilizada.

Quando uma página vai ser substituída, o algoritmo escolhe a página que não foi utilizada durante o maior período de tempo.

Como podemos perceber, cada vez que ocorre uma referência a uma página, é necessário uma atualização no tempo associado a ela, o que introduz um atraso na execução do processo. Além disso, a escolha da página a ser substituída deve ser feita com uma pesquisa em todas as páginas para se determinar aquela com o menor tempo associado. Apesar desse esforço adicional, esse método tem resultados melhores do que o primeiro.

Existem outros algoritmos mais simples de serem implementados que se aproximam da performance obtida pelo LRU e podem ser encontrados nas referências.

O processo do sistema operacional que chamamos de carregador, para todos os programas, carrega na memória a tabela

de páginas e a primeira página. Após isso, uma nova página só é carregada quando pedida através de uma falta. Essa política de carregamento é conhecida por paginação sob demanda.

Os melhoramentos aqui propostos podem ser feitos no sistema operacional implementado, de forma a se estudar o desempenho do mesmo em cada nova situação. Esse é o objetivo principal desse trabalho.

Um dos pontos mais críticos no projeto de um sistema operacional, está relacionado ao bloqueio perpétuo ("Deadlock"). O estado de bloqueio perpétuo ocorre quando dois ou mais processos estão suspensos esperando por um recurso que só pode ser liberado por um desses processos. Um exemplo típico dessa situação foi discutido no capítulo 3, no controle de alocação de acopladores.

Basicamente, esse problema pode ser tratado com duas abordagens diferentes. A primeira, que foi usada por nós, e é normalmente adotada, utiliza um protocolo que garante que o sistema nunca entre em bloqueio perpétuo. A segunda abordagem, é permitir que o sistema entre em bloqueio e então recuperá-lo.

Um método frequentemente adotado na primeira abordagem é fazer a alocação dos recursos necessários para um processo, antes do início de sua execução, dessa forma, um processo inicia sua execução com a garantia de que não será bloqueado. A restrição colocada a esse método, é de que, como os recursos são todos previamente alocados, eles podem permanecer sem serem usados por um período de tempo razoavelmente longo, no qual poderiam estar sendo utilizados por outros processos.

No nosso sistema, o método adotado permite que os recursos sejam alocados à medida que os processos progridem. Por exemplo, suponha três processos chamados P1, P2 e P3. O processo P1 necessita para ser totalmente executado dos recursos R1, R3, R5, e R7; o processo P2 necessita dos recursos R2, R3, R5, e R7; o processo P3 de R3, R4, R5, e R6. O processo P1 iniciando sua execução, ganha o recurso R1, o processo P2 inicia e ganha o recurso R2, e o processo P3 ganha o recurso R3. Nesse instante, o processo P1 não pode progredir, já que R3, não está disponível, o mesmo ocorre com P2. Somente P3 pode continuar e

ganha dessa forma os recursos R4, R5, e R6. Ao término de sua execução, os recursos são liberados, permitindo então que P1 progrida e depois P2.

Nesse exemplo, foi feita uma suposição nem sempre válida, de que a liberação dos recursos obtidos por P3 é feita somente no final de sua execução. Um recurso pode ser liberado assim que sua utilização não for mais necessária, podendo então ser alocado para outro processo. No caso citado, se o recurso R3 for liberado por P3 antes do seu término, o processo P1 poderá ser disparado, já que foi suspenso por necessitar desse recurso e, assim que P1 liberá-lo, P2 também poderá ser disparado.

A sequência de progressão de cada um dos processos do nosso sistema, exclue a possibilidade de ocorrência de bloqueio perpétuo.

Esse método é discutido em Segre e Kirner [25], analisando o problema de bloqueio perpétuo em chamadas aninhadas de monitores.

Na figura 4.1 abaixo, mostramos um resumo das principais alternativas de implementação dos diferentes processos do Sistema Operacional.

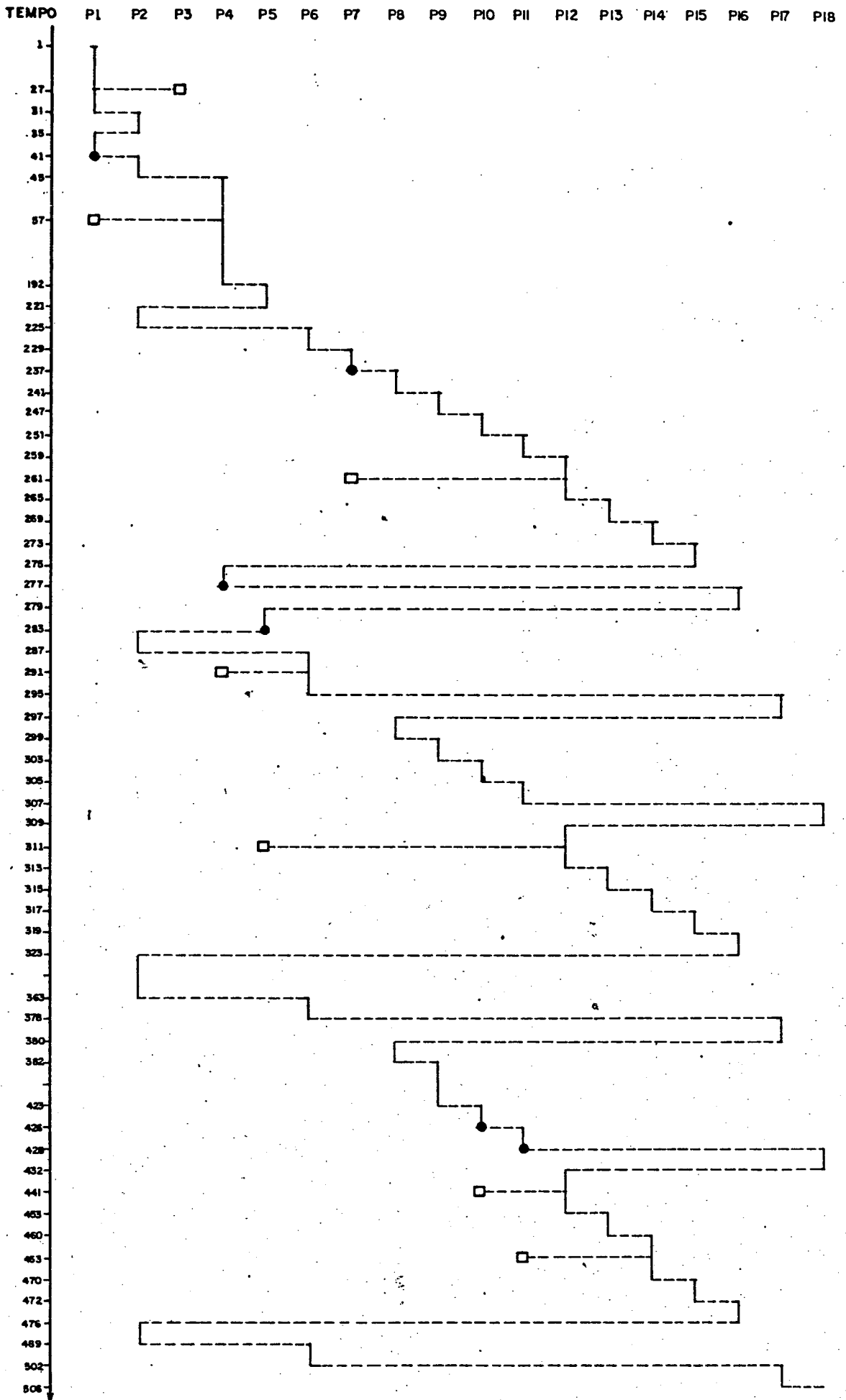
PROCESSO	POLÍTICA ADOTADA	ALTERNATIVAS
Agendador	Algoritmo "ROUND-ROBIN"	Fila de usuários com prioridades de execução
Paginação	FIFO	LRU, OPT, Segunda chance e outros

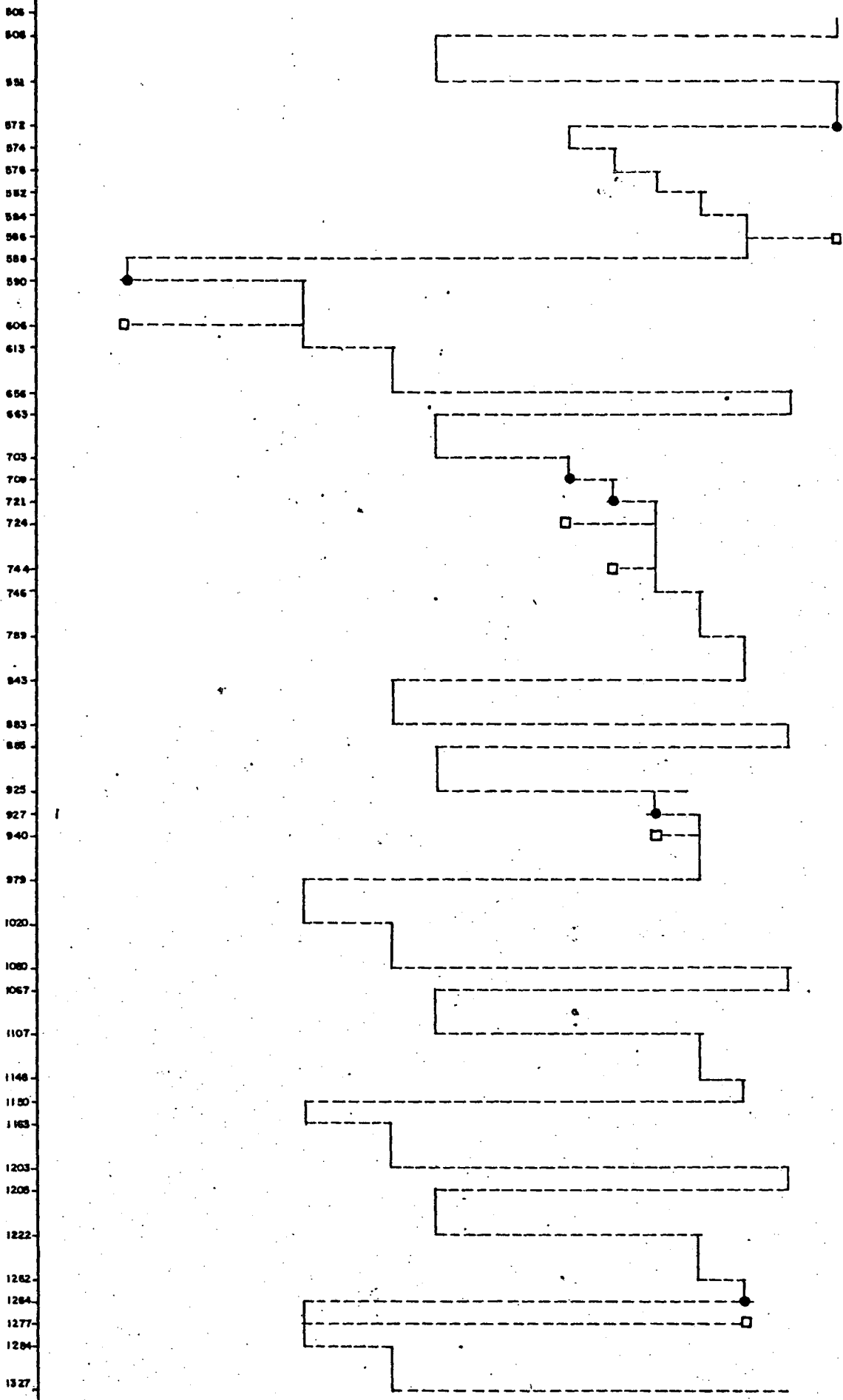
FIG.4.1 - RESUMO DAS PRINCIPAIS ALTERNATIVAS DE IMPLEMENTAÇÃO

A listagem mostrada no apêndice dois, é o resultado da execução pelo sistema de 18 programas de usuários. Nela procuramos mostrar o compartilhamento da UCP entre os processos do sistema operacional e os usuários, de tal forma que, estando um processo de usuário em execução e sendo interrompido, é mostrada a causa da interrupção, o tempo que ele permaneceu executando, e a ação seguinte tomada pelo sistema.

Aqui, vamos mostrar um resumo dessa execução através de um gráfico de tempo. Na figura 5.1 apresentamos esse gráfico e nele adotamos as seguintes convenções:

- 1 - ● fim da execução
- 2 - □ impressão do programa





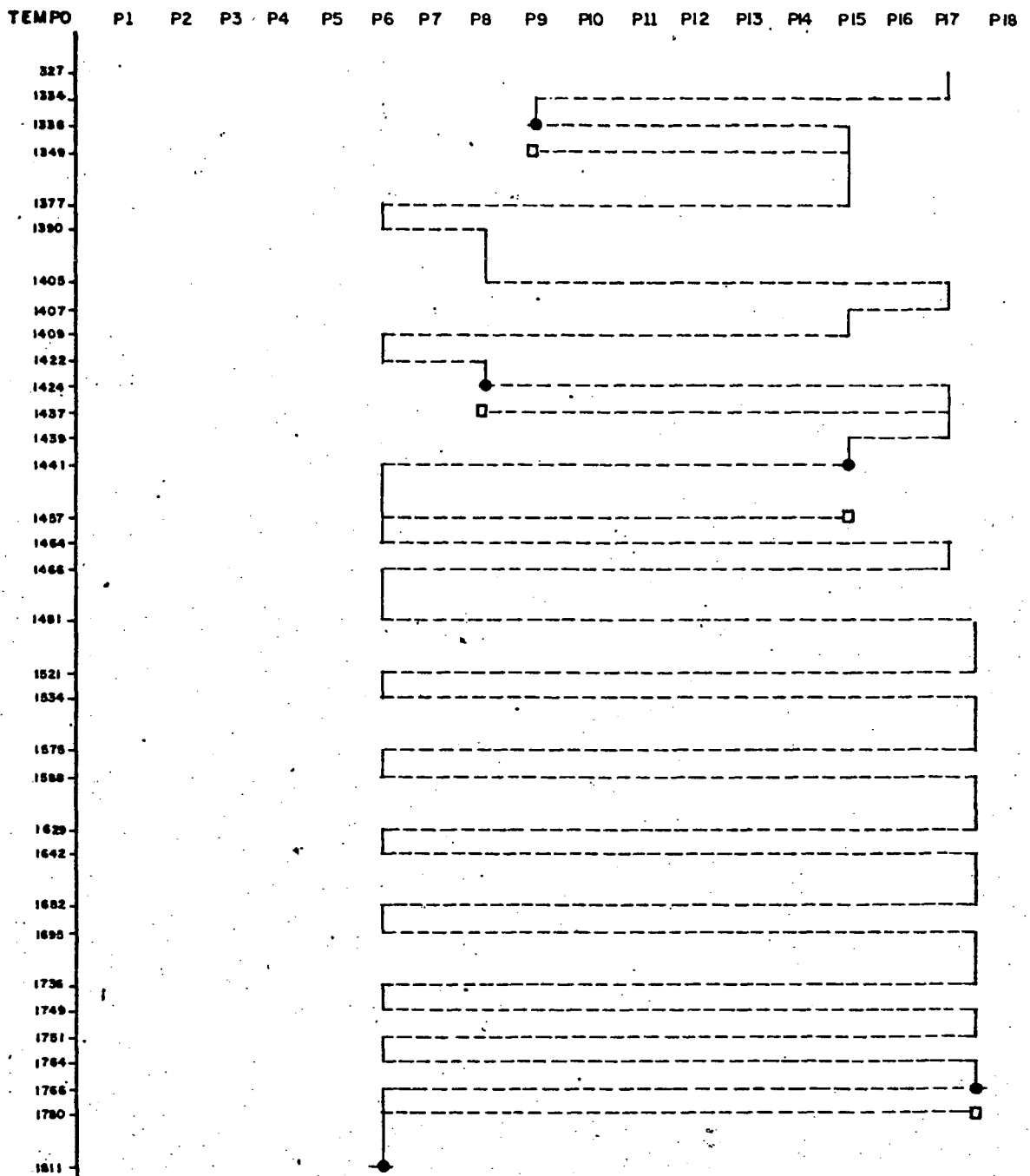


FIG. 5.1 - COMPARTILHAMENTO DA UCP ENTRE OS USUÁRIOS

Analisando a figura, podemos notar por exemplo, que no tempo 1, iniciou-se a execução do programa do usuário 1, no tempo 27 foi fornecido na impressora o programa do usuário 3, que não chegou a ser processado. Esse programa foi submetido com erro nos cartões de controle. No tempo 31 foi interrompida a execução do programa do usuário 1 e iniciada a execução do programa do usuário 2, que foi interrompida no tempo 35, voltando a UCP a executar o programa do usuário 1. No tempo 41, terminou a execução do programa 1 e reiniciou a execução do programa 2, e assim por diante.

Por esse gráfico, podemos perceber claramente a troca na UCP dos programas em execução.

Todos os detalhes a respeito de como foram escritos os programas, o que cada um faz, também estão no apêndice 2.

Dentro do objetivo para o qual esse trabalho foi desenvolvido, ou seja, para aplicações didáticas, acreditamos que da forma como está estruturado, facilitará bastante o entendimento da estrutura e funcionamento de sistemas operacionais.

Uma vez apresentados os conceitos pertinentes, damos ao aluno uma ferramenta na qual esses conceitos podem ser aplicados, ou seja, estamos fornecendo um sistema que ele poderá alterar a partir de alternativas existentes para a implementação de cada processo que o compõe como discutido no capítulo 4.

Dessa forma, poderá ser estudado o desempenho do sistema operacional face a cada alteração executada, e ainda, as consequências que tal alteração pode trazer ao sistema como um todo.

Com a experiência adquirida a partir desse estudo, acreditamos ser mais fácil entender os sistemas operacionais reais com os quais ele deverá trabalhar.

B I B L I O G R A F I A

- [1] APPELBE, F.W., HANSEN, K. - "A Survey of System Programming Languages: Concepts and Facilities". Software - Practice and Experiences, vol. 15, 1985.

- [2] BRINCH HANSEN, P. - "Operating System Principles". Prentice Hall, Englewood Cliffs, New Jersey, 1970.

- [3] BRINCH HANSEN, P. - "The Architecture of Concurrent Programs". Prentice Hall, Englewood Cliffs, New Jersey, 1977.

- [4] BRINCH HANSEN, P. - "The TRIO Operating System". Software - Practice and Experience, vol. 10, 1981.

- [5] BRINCH HANSEN, P. - "Edison - a Multiprocessor Language". Software - Practice and Experience, vol. 11, 1981.

- [6] CHU, Y. - "Computer Organization and Microprogramming". Mc Graw - Hill Book Co., New York, 1962.

- [7] GUIMARÃES, C.C. - "Princípios de Sistemas Operacionais". Campus, MEC/SEPLAN, Rio de Janeiro, 1980.

- [8] HOARE, C.A.R. - "Monitors: An Operating System Structuring Concept". Communications of ACM, vol. 17, 1974.



- [9] HOARE, C.A.R. - "Communicating Sequential Processes" .
Communications of ACM, vol. 21, 1978.
- [10] HOLT, R.C., WORTMAN, D.B., BARNARD, D.T., CORDY, J.R. -
"SP/K: A System for Teaching Computer Programming". Co-
mmunications of ACM, vol. 5, 1977.
- [11] HOLT, R.C., LAZOWSKA, E.D., GRAHAM, G.S., SCOTT, M. A. -
"Structured Concurrent Programming with Operating Sys-
tems Applications". Addison - Wesley, Reading, Massachu-
setts, 1978.
- [12] KAISLER, S.H. - "The Desing of Operating Systems for
Small Computer Systems". John Wiley & Sons, New York,
1983.
- [13] KERNINGHAN, W.B., RITCHIE, M.D. -"The C Programming Lan-
guage". Prentice Hall, Englewood Cliffs, New Jersey,
1978.
- [14] KIRNER, C., SEGRE, L., SANTOS, S, M. - "Sistema Operacio-
nal Didático: Caracterização e Experiências". Anais da
Conferência Latino Americana de Informática, Lima, Peru,
1982.
- [15] KRUIJER, H.S.M. - "A Multi-user Operating System for
Transaction Processing Written in Concurrent Pascal " .
Software - Practice and Experience, vol. 12, 1982.
- [16] LANGDON Jr., G.G., FREGNI, E. - "Projeto de Computadores
Digitais". Edgar Blucher Ltda, São Paulo, 1977.

- [17] LORIN, H., DEITEL, H. - "Operating Systems". Addison Wesley, Reading, Massachusetts, 1981.
- [18] MCKENDRY, M.S. - "The Use of Monitors in Microprocessor Software Development". Euromicro Journal, vol. 4, 1978.
- [19] PETERSON, J.L., SILBERSCHATZ, A. - "Operating System Concepts". Addison Wesley, Reading, Massachusetts, 1983.
- [20] RAMALHO, F.E.R. - "Software Básico". MEC, Departamento - de Assuntos Universitários, 1973.
- [21] RITCHIE, M.D., THOMPSON, K. - "The UNIX Time-Sharing Systems". Communications of ACM, vol. 17, 1974.
- [22] ROBERT Fº, E., BARBOSA, J.A. - "Sistemas Operacionais" . Notas de Aula, 1974.
- [23] SEGRE, L. - "Mecanismos de Comunicação para Processos Concorrentes". Relatório Técnico, COPPE/UFRJ, 1981.
- [24] SEGRE, L., SANTOS, S.M. - "O Conceito de Monitor como Instrumento de Sincronização em Programação Concorrente" Relatório Técnico, COPPE/UFRJ, 1981.
- [25] SEGRE, L. KIRNER, C. - "Primitivas de Comunicação e Sincronização de Processos por Troca de Mensagens: Uma Análise". Relatório Técnico, DCES/UFSCar, 1982.

- [26] SMITH, A., BELL, D.H. - "A Multiprogramming System for a Microcomputer". Software - Practice and Experience, , vol. 12, 1982.
- [27] WIRTH, N. - "The Programming Language Pascal". Acta Informatica, vol. 1, 1971.
- [28] WIRTH, N. - "Modula: a Language for Modular Multiprogramming". Software - Practice and Experience, vol. 7, 1977.
- [29] ZUFFO, J.A. - "Sistemas Eletrônicos Digitais - Organização Interna e Projeto". Edgar Blucher, S. Paulo, vol. 2, 1976.

APÉNDICE

I

REPERTÓRIO DE INSTRUÇÕES

As instruções que compõe o repertório, foram divididas em quatro grupos. As que fazem referência à memória, as que fazem referência à registro, as que usam endereçamento imediato, e as que usam endereçamento especial.

O usuário fornece sua instrução, obedecendo à seguinte ordem:

código da operação, modo de endereçamento, (registro). O registro foi colocado entre parênteses, já que pode não estar presente em algumas instruções. Nas instruções que serão apresentadas, usaremos as seguintes abreviações:

- R para registro;
- ME para modo de endereçamento;
- D para deslocamento;
- N para número de rotações.

As instruções que fazem referência à memória, estão listadas a seguir, com a explicação de seu significado. Cabe ressaltar, que o modo de endereçamento utilizado por essas instruções, podem ser: Direto (DI), Indexado 1 (I1), Indexado 2 (I2), e relativo (RL).

- | | |
|--------------------|--|
| 1-) JMP, ME
Mem | ; instrução de desvio incondicional, sendo que o endereço que serve como referência , está na 2 ^a palavra. |
| 2-) PUSH, DI, R | ; instrução que coloca o conteúdo do registrador R no topo da pilha. É usada sempre com endereçamento direto. |
| 3-) BRA, DI
D | ; instrução de desvio incondicional, na 2 ^a palavra, temos o deslocamento, e é sempre executada com endereçamento direto. |

- 4-) POP, DI, R ; instrução que coloca o conteúdo do topo da pilha no registrador R. Usada sempre com endereçamento direto.
- 5-) JSR, ME ; instrução de desvio para subrotina. Na 2ª palavra temos o endereço de referência.
Mem
- 6-) BMI, ME ; desvio condicional, executado se o resultado de instrução executada anteriormente a essa, gerou um valor negativo.
Mem
- 7-) BEQ, ME ; desvio condicional, executado se o resultado de instrução anterior gerou um valor igual a zero.
Mem
- 8-) BCC, ME ; desvio condicional executado se o registrador C ("carry") é igual a zero.
Mem
- 9-) BVS, ME ; desvio condicional executado se o resultado de instrução anterior a essa deu "overflow".
Mem
- 10-) STO, ME, R ; carrega a posição de memória com o conteúdo do registrador R.
Mem
- 11-) LDD, ME, R ; carrega o registrador R com o conteúdo de memória.
Mem
- 12-) INC, ME ; incrementa posição de memória.
Mem
- 13-) DCR, ME ; decrementa posição de memória.
Mem
- 14-) CLR, ME ; zera posição de memória.
Mem

- 15-) ADD, ME, R
Mem ; soma o conteúdo do registrador R, com o conteúdo da posição de memória. O resultado fica armazenado no registrador.
- 16-) SUB, ME, R
Mem ; subtração com efeito igual a 15.
- 17-) CMP, ME, R
Mem ; compara registrador R com posição de memória.
- 18-) COM, ME
Mem ; complementa posição de memória.
- 19-) AND, ME, R
Mem ; operação lógica "E", entre registro R e posição de memória.
- 20-) EOR, ME, R
Mem ; operação lógica "ou-exclusivo", entre o registro R e posição de memória.
- 21-) IOR, ME, R
Mem ; operação lógica "OU", entre o registro R e posição de memória.
- 22-) ROL, ME
Mem ; deslocamento à esquerda na posição de memória.
- 23-) ROR, ME
Mem ; deslocamento à direita na posição de memória ("rotate right").
- 24-) ASL, ME
Mem ; deslocamento aritmético à esquerda ("shift left").
- 25-) ASR, ME
Mem ; deslocamento aritmético à direita ("shift right").
- 26-) LSR, ME
Mem ; deslocamento lógico



Algumas das instruções com referência a registro são similares às instruções acima, e essas não serão explicadas. O modo de endereçamento das instruções a seguir é o de registro (RE).

1-) INC, RE, r

2-) DCR, RE, R

3-) CLR, RE, R

4-) COM, RE, R

5-) ADD, RE, R

R

6-) SUB, RE, R

R

7-) CMP, ME, R

R

8-) AND, RE, R

R

9-) EOR, RE, R

R

10-) IOR, RE, R

R

11-) ROL, RE, R

D

; na segunda palavra temos o deslocamento (número de vezes).

12-) ROR, RE, R

D

13-) ASL, RE, R

D

14-) ASR, RE, R

D

15-) LSR, RE, R

D

- 16-) TFR, RE, R ; transfere o conteúdo de re-
R gistrador da segunda palavra
para o indicado na primeira.
- 17-) CLC, RE ; zera o registrador C ("carry
").
- 18-) CLI, RE ; zera o regsitrador de inter-
rupção CNTR.
- 19- SEC, RE ; ativa o registrador C.
- 20-) SEI, RE ; ativa o registrador de inter-
rupção CNTR.

As instruções com endereçamento imediato, são as mes-
mas numeradas de 10 à 26 nas instruções com referência à memóri-
a, trocando o modo de endereçamento por imediato(IM), e a se-
gunda palavra pelo próprio operando.

Finalmente as instruções com modo de endereçamento es-
pecial (ES), são:

- 1-) NOP, ES ; nenhuma operação.
- 2-) FIM, ES ; fim de programa.
- 3-) RTS, ES ; retorno de subrotina.
- 4-) LER, ES, R ; lê o dado e o coloca no re-
gistrador R.
- 5-) IMP, ES, R ; imprime o conteúdo do regis-
trador R.

Os possíveis registradores (R) que podem ser utiliza-
dos nas instruções são: A, B, IX1, IX2.

A listagem mostrada nesse apêndice, é o resultado da execução pelo sistema, de 18 programas de usuários. Nessa listagem, procuramos mostrar o compartilhamento da UCP entre os processos do sistema e dos usuários. Dessa forma, estando um processo de usuário em execução, e sendo interrompido, é mostrada a causa de sua interrupção, além do tempo em que ele permaneceu em execução pela UCP.

A entrada dos programas foi fornecida em ordem crescente do número do usuário.

Abaixo, damos uma breve descrição de cada um dos programas submetidos para execução:

Usuário 1: Soma entre dois valores fornecidos, através de endereçamento imediato.

Usuário 2: Multiplicação através de somas sucessivas. Os dois primeiros valores impressos, foram fornecidos pelo usuário, e o terceiro, é o resultado.

Usuário 3: O usuário ao fornecer o programa, não colocou o cartão de controle de início de programa.

Usuário 4: Subtração de dois valores, fornecidos através de endereçamento imediato.

Usuário 5: Soma de dois valores, com inibição de erro.

Usuário 6: Multiplicação através de somas sucessivas, com impressão de todos os resultados intermediários.

Usuário 7: O usuário, não forneceu os dados necessários, para a execução de seu programa.

Usuário 8: Idem ao programa do usuário 2.

Usuário 9: Cálculo do fatorial de um número. O primeiro valor impresso, é o número fornecido pelo usuário, e o segundo, o fatorial calculado.

Usuário 10: O usuário no seu programa, tenta acessar posição fora dos limites do mesmo.

- Usuário 11: O usuário no seu programa, escreveu um código de operação inexistente no repertório de instrução.
- Usuário 12: Soma de dois valores cujo resultado deu "overflow". Os dois valores impressos foram os fornecidos pelo usuário.
- Usuário 14: Cálculo da expressão $Z = 2*(X - Y) - X$. O primeiro valor impresso, corresponde a X, o segundo a Y, e o terceiro é o resultado correspondente a Z.
- Usuário 15: Divisão de dois valores. O primeiro valor impresso, corresponde ao dividendo, o segundo é o divisor, e o terceiro é o quociente.
- Usuário 16: Idem ao 15, com o quarto valor impresso correspondendo ao resto da divisão. O valor correspondente ao resto da divisão no programa do usuário 15 não foi impresso, porque a divisão dos valores fornecidos pelo usuário, deu resto igual a zero.
- Usuário 17: Média aritmética dos 5 primeiros números impressos. O sexto valor corresponde ao resultado.
- Usuário 18: Cálculo do fatorial. O valor impresso, corresponde ao número fornecido pelo usuário. O tempo dimensionado para execução do programa foi insuficiente.

Para facilitar a visualização dos resultados fornecidos após a execução desse lote de programas, temos a seguir, qual seria a sequência de saída dos mesmos na impressora.

Já vimos que, nesse sistema, quando a UCP entra em estado de espera, é porque não temos nem processo do sistema pronto para rodar e nem processo de usuário. Nesse estado, somente uma interrupção de periférico reativa o funcionamento da mesma.

A mensagem "Término do Sistema" no fim da listagem, significa que a UCP entrou em estado de espera, e que todos os periféricos estão desligados, não tendo portanto mais nenhum serviço para ser executado.

APÉNDICE

II

PROGRAMA DO USUARIO 2 CARTAO DE PROGRAMA FORA DO LUGAR

IMPRESSAO DO PROGRAMA DO USUARIO 1

CLC,RE
CLI,RE
LCC,IM,A
16000
ADD,IM,A
12000
IMP,ES,A
FIM,ES

IMPRESSAO DOS RESULTADOS

28000

PROGRAMA DO USUARIO 1 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 7

LER,ES,A
CLC,RE
CLI,RE
ADD,IM,A
1800
IMP,ES,A
FIM,ES

IMPRESSAO DOS RESULTADOS

PROGRAMA DO USUARIO 7 TENTATIVA DE LER/ESCREVER

APOS TER ACABADO DADOS OU PAGINA DE IMPRESSAO

IMPRESSAO DO PROGRAMA DO USUARIO 4
CLC, PE
CLI, RE
LDC, IM, A
T8000
SUB, IM, A
14237
IMP, ES, A
FIM, ES

IMPRESSAO DOS RESULTADOS
3763

PROGRAMA DO USUARIO 4 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 5
CLC, RE
SEI, PE
LDD, IM, A
32767
ADD, IM, A
12000
IMP, ES, A
FIM, ES

IMPRESSAO DOS RESULTADOS
44767

PROGRAMA DO USUARIO 5 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 10

LER, ES, A

IMP, ES, A

STO, DI, A

37

LER, ES, E

AND, CI, E

37

IMP, ES, E

FIM, ES

IMPRESSAO DOS RESULTADOS

20

PROGRAMA DO USUARIO 10 PROTECAO MEMORIA

IMPRESSAO DO PROGRAMA DO USUARIO 11

LER, ES, A

IMP, ES, A

***, DI, A

11

LSR, DI

11

LDC, CI, A

11

IMP, ES, A

FIM, ES

IMPRESSAO DOS RESULTADOS

1024

PROGRAMA DO USUARIO 11 COD.OP. INVALIDO

IMPRESSAC DO PROGRAMA DO USUARIO 18

LEF, ES, A

IMP, ES, A

STC, DI, A

29

TFR, RE, B

1

DCR, RE, E

BFC, DI

27

STC, DI, E

30

DCR, RE, B

BFC, DI

27

ACC, CI, A

29

DCR, RE, E

BFC, DI

21

BRA, CI

14

LDD, DI, E

30

STC, DI, A

29

BRA, DI

6

IMP, ES, A

FIM, ES

IMPRESSAC DOS RESULTADOS

16

PROGRAMA, DO USUARIO 18 TEMPO ESGOTADO

IMPRESSAC DO PROGRAMA DO USUARIO 2

LFR, ES, A

IMP, ES, A

LER, ES, E

IMP, ES, E

STC, DI, A

18

DCR, RE, E

BEQ, CI

16

ADD, DI, A

18

DCR, RE, E

BEQ, CI

16

BRA, CI

C

IMP, ES, A

FIM, ES

IMPRESSAC DOS RESULTADOS

10

5

50

PROGRAMA DO USUARIO 2 FIM NORMAL

IMPRESSAC DO PROGRAMA DO USUARIO 12

LER, ES, A

IMP, ES, A

STC, DI, A

10

LER, ES, A

IMP, ES, A

ADD, DI, A

10

IMP, ES, A

FIM, ES

IMPRESSAC DOS RESULTADOS

32737

5210

PROGRAMA DO USUARIO 12 OVERFLOW

IMPRESSAO DO PROGRAMA DO USUARIO 13

LER, ES, B

IMP, FS, E

STO, DI, E

10

LER, ES, E

IMP, ES, B

SUB, DI, B

10

IMP, FS, B

FIM, FS

IMPRESSAO DOS RESULTADOS

10

-32767

PROGRAMA DO USUARIO 13 UNDERFLOW

IMPRESSAO DO PROGRAMA DO USUARIO 14

LER, ES, A

IMP, ES, A

STO, DI, A

14

LER, ES, B

IMP, ES, B

SUB, RE, A

2

ACC, RE, A

1

SUB, DI, A

14

IMP, ES, A

FIM, ES

IMPRESSAO DOS RESULTADOS

50

12

26

PROGRAMA DO USUARIO 14 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 16

LER, ES, A
IMP, ES, A
LER, ES, B
IMP, ES, B
CLM, DI
27
SUB, RE, A
2
BEQ, CI
20
BMI, DI
16
INM, DI
27
BRA, CI
6
ADD, RE, A
2
BRA, DI
22
INM, DI
27
LDD, DI, B
27
IMP, ES, B
IMP, ES, A
FIM, ES

IMPRESSAO DOS RESULTADOS

15
4
3
3

PROGRAMA DO USUARIO 16 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 9

LER, ES, A

IMP, ES, A

STC, DI, A

29

TFR, RE, E

1

DCR, RF, E

BEC, DI

27

STO, DI, E

30

DCR, RE, E

BEC, DI

27

ADC, DI, A

29

CCR, RE, E

BEC, DI

21

BRA, DI

14

LDD, DI, E

30

STC, DI, A

29

BRA, DI

6

IMP, ES, A

FIM, ES

IMPRESSAO DOS RESULTADOS

6

720

PROGRAMA DO USUARIO 9 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 8

LER,ES,A

IMP,ES,A

LER,ES,E

IMP,ES,E

STO,DI,A

18

DCR,RE,B

BEG,CI

16

ADD,DI,A

18

DCR,RE,E

BEG,CI

16

ERA,CI

9

IMP,ES,A

FIM,ES

IMPRESSAO DOS RESULTADOS

10

20

200

PROGRAMA DO USUARIO 8 FIM NORMAL

IMPRESSAO DO PROGRAMA DO USUARIO 15

LER, ES, A

IMP, ES, A

LER, ES, B

IMP, ES, B

CLM, DI

27

SUB, RE, A

2

BEG, DI

20

BMI, DI

16

INM, DI

27

BRA, DI

6

ADD, RE, A

2

BRA, DI

22

INM, DI

27

LDD, DI, B

27

IMP, ES, B

IMP, ES, A

FIM, ES

IMPRESSAO DOS RESULTADOS

144

12

12

PROGRAMA DO USUARIO 15 FIM NORMAL



IMPRESSAO DO PROGRAMA DO USUARIO 17

LER, ES, A

IMP, FS, A

STO, DI, A

50

LER, ES, A

IMP, ES, A

STC, DI, A

51

LER, FS, A

IMP, ES, A

STO, DI, A

52

LER, ES, A

IMP, ES, A

STO, DI, A

53

LER, ES, A

IMP, ES, A

STC, DI, A

54

LDD, DI, A

50

ADD, DI, A

51

ADD, DI, A

52

ADD, DI, A

53

ADD, DI, A

54

LDD, IN, E

5

CLM, DI

55

SUB, RE, A

2

BEC, DI

44

BMI, DI

46

INM, DI

55

BRA, DI

34

INM, DI

55

LDD, DI, E

55

IMP, ES, E

FIN, ES

IMPRESSAO DOS RESULTADOS

5
10
6
70
14
11

PROGRAMA DO USUARIO 17 FIM NORMAL

PASCAL 8000/1.2 AAEC (1ST FEB 78)

31/05/85 AT 17:17:07 PAGE 1

C620 -- PROGRAM SO(INPUT,OUTPUT)

*AAEC PASCAL COMPILATION CONCLUDED *

*NO ERRORS DETECTED IN PASCAL PROGRAM *

SAIDA NA IMPRESSORA DOS* ACAO SOBRE OS PROGRAMAS DOS USUARIOS * PROCESSOS DO SISTEMA OPERACIONAL
PROGRAMAS DOS USUARIOS * * *

RELOGIO DO SISTEMA= 1

ENTROU NO PROCESSO DE LEITURA
UCP PARADA ESPERANDO INTERRUPCAO DE PERIFERICO
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 1
ENTROU NO PROCESSO DE LEITURA
UCP PARADA ESPERANDO INTERRUPCAO DE PERIFERICO
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 1
UCP PARADA ESPERANDO INTERRUPCAO DE PERIFERICO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 1

RELOGIO DO SISTEMA= 3

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 5

ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 2
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 5

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 8

RELOGIO DO SISTEMA= 8

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 11

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 2

RELOGIO DO SISTEMA= 11

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 14

RELOGIO DO SISTEMA= 14

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 17

ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 3
ENTROU NO PROCESSO DE LEITURA
ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 3
ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 17

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 27

IMPRESSAO DOS RESULTADOS

PROGRAMA CC USUARIO 3 CARTAO DE PROGRAMA FCRA DO LUGAR

```
*****
RELOGIO DO SISTEMA= 27
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 28
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 4
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 29
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 1
RELOGIO DO SISTEMA= 31
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 2
*****
RELOGIO DO SISTEMA= 31
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 32
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 4
*****
RELOGIO DO SISTEMA= 33
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 35
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 1
*****
RELOGIO DO SISTEMA= 35
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 37
*****
RELOGIO DO SISTEMA= 37
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 39
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 5
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 39
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 1
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 1
RELOGIO DO SISTEMA= 41
CAUSA DA SUSPENSAC FOI FIM OU ERRO

ENTROU NO SPOOL DE SAIDA
```

SPDOL DE SAIDA DO PROG. DO USUARIO 1
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 2

RELOGIO DO SISTEMA= 41
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 42

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 5

RELOGIO DO SISTEMA= 43
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 45
CAUSA DA SUSPENSAC FCI IASTRUCAC DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 4

RELOGIO DO SISTEMA= 45
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 47

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 47
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 49

ENTROU NO SPDOL DE ENTRADA
SPDOL DE ENTRADA DO PROGRAMA DO USUARIO 6
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 49
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 51

RELOGIO DO SISTEMA= 51
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 54

RELOGIO DO SISTEMA= 54
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 57

IMPRESSAO DO PROGRAMA DO USUARIO 1
CLF, PE
CLT, PE
LDC, IM, A
16000
SEC, IM, A
12000
IMP, ES, A
FTM, ES

ENTROU NO PROCESSO DE IMPRESSAO

```
*****
RELOGIO DO SISTEMA= 57
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 60
```

```
*****
ENTROU NO PROCESSO DE LEITURA
```

```
*****
RELOGIO DO SISTEMA= 60
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 63
```

```
*****
RELOGIO DO SISTEMA= 63
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 66
```

```
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 6
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 7
```

```
*****
RELOGIO DO SISTEMA= 66
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 69
```

1
88
1
IMPRESSAO DOS RESULTADOS
28000

PROGRAMA DO USUARIO 1 FIM NORMAL

```
*****
RELOGIO DO SISTEMA= 69
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 72
```

```
*****
ENTROU NO PROCESSO DE LEITURA
```

```
*****
RELOGIO DO SISTEMA= 72
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 75
```

```
*****
RELOGIO DO SISTEMA= 75
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 78
```

```
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 7
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 8
```

```
*****
RELOGIO DO SISTEMA= 78
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 81
```

```
*****
RELOGIO DO SISTEMA= 81
```



```

VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 84
*****
ENTROU NO PROCESSO DE LFITURA
*****
RELOGIO DO SISTEMA= 84
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 87
*****
RELOGIO DO SISTEMA= 87
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 90
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 8
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 9
*****
RELOGIO DO SISTEMA= 90
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 93
*****
RELOGIO DO SISTEMA= 93
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 96
*****
ENTROU NO PROCESSO DE LFITURA
*****
RELOGIO DO SISTEMA= 96
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 99
*****
ENTROU NO SPOOL DE ENTRADA
*****
RELOGIO DO SISTEMA= 99
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 102
*****
RELOGIO DO SISTEMA= 102
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 105
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 8
*****
RELOGIO DO SISTEMA= 105
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 108
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 10
ENTROU NO PROCESSO DE LFITURA
*****
RELOGIO DO SISTEMA= 108

```

```

VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 111
*****
PELOGIO DO SISTEMA= 111
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 114
*****
RELOGIO DO SISTEMA= 114
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 117
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 11
*****
RELOGIO DO SISTEMA= 117
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 120
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 11
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 120
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 123
*****
RELOGIO DO SISTEMA= 123
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 126
*****
RELOGIO DO SISTEMA= 126
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 129
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 11
*****
RELOGIO DO SISTEMA= 129
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 132
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 12
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 132
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA= 135
*****
RELOGIO DO SISTEMA= 135
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO

```

```
RELOGIO DO SISTEMA= 13F
*****
RELOGIO DO SISTEMA= 13P
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 14J
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 12
*****
RELOGIO DO SISTEMA= 14I
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 144
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 13
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 144
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 147
*****
RELOGIO DO SISTEMA= 147
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 150
*****
RELOGIO DO SISTEMA= 150
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 153
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 12
*****
RELOGIO DO SISTEMA= 153
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 156
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 14
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 156
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 159
*****
RELOGIO DO SISTEMA= 159
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 162
*****
RELOGIO DO SISTEMA= 162
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 165
*****
```

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 14

RELOGIO DO SISTEMA= 165
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 168

ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 15
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 168
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 171

RELOGIO DO SISTEMA= 171
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 174

RELOGIO DO SISTEMA= 174
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 179

ENTROU NO SPOOL DE ENTRADA
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 179
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 182

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 182
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 185

RELOGIO DO SISTEMA= 185
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 190

ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 16
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 190
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4

ENTROU NO COMPLEMENTO INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 4
RELOGIO DO SISTEMA= 192
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 5

```

RELOGIO DO SISTEMA= 192
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 194
*****
RELOGIO DO SISTEMA= 194
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 201
*****
ENTROU NO SPOOL DE ENTRADA
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 201
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 204
*****
ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 16
*****
RELOGIO DO SISTEMA= 204
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 207
*****
RELOGIO DO SISTEMA= 207
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 212
*****
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 17
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 212
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 215
*****
RELOGIO DO SISTEMA= 215
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
*****
ENTROU NO COMPLEMENTA INTERRUPCAO
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 5
RELOGIO DO SISTEMA= 221
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 2
*****
RELOGIO DO SISTEMA= 221
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 223
*****
ENTROU NO SPOOL DE ENTRADA
ENTROU NO PROCESSO DE LEITURA
*****
RELOGIO DO SISTEMA= 223
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 225
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA= 225
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 227

RELOGIO DO SISTEMA= 227
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 229
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 7

RELOGIO DO SISTEMA= 229
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 7
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTO DE PERIFERICO
RELOGIO DO SISTEMA= 231

RELOGIO DO SISTEMA= 231
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 7
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTO DE PERIFERICO
RELOGIO DO SISTEMA= 233

ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 233
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 7
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTO DE PERIFERICO
RELOGIO DO SISTEMA= 235

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 17
ENTROU NO SPOOL DE ENTRADA
SPOOL DE ENTRADA DO PROGRAMA DO USUARIO 18

RELOGIO DO SISTEMA= 235
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 7

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 7
RELOGIO DO SISTEMA= 237
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8
ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 7

RELOGIO DO SISTEMA= 237
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTO DE PERIFERICO
RELOGIO DO SISTEMA= 239

```

RELOGIO DO SISTEMA= 239
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 8
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 241
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALDU PARA RODAR PROG. DO USUARIO 9
*****
RELOGIO DO SISTEMA= 241
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 9
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 243
*****
ENTROU NO PROCESSO DE LEITURA

RELOGIO DO SISTEMA= 243
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 9
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 245
*****
ENTROU NO SPOOL DE ENTRADA

RELOGIO DO SISTEMA= 245
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 9
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 247
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO ESCALADOR
ESCALDU PARA RODAR PROG. DO USUARIO 10
*****
RELOGIO DO SISTEMA= 247
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 10
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 249
*****
ENTROU NO PROCESSO DE IMPRESSAO
ENTROU NO PROCESSO DE E/S USUARIO

RELOGIO DO SISTEMA= 249
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 10
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 10
RELOGIO DO SISTEMA= 251
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO ESCALADOR
ESCALDU PARA RODAR PROG. DO USUARIO 11
*****
RELOGIO DO SISTEMA= 251
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 11
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 253
*****
RELOGIO DO SISTEMA= 253
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 11
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 255
*****
ENTROU NO PROCESSO DE E/S USUARIO
*****

```

RELOGIO DO SISTEMA= 255
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 11

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 11
RELOGIO DO SISTEMA= 257
CAUSA DA SUSPENSAC FCI INSTRUCAO DE E/S

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 12

RELOGIO DO SISTEMA= 257
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 259

IMPRESSAO DO PROGRAMA DO USUARIO 7

LEF,FS,A
CLC,PE
CLI,PE
ADC,TR,A
JPCO
IMP,ES,A
FIM,ES

RELOGIO DO SISTEMA= 259
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 261

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 261
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 12
RELOGIO DO SISTEMA= 263
CAUSA DA SUSPENSAC FCI INSTRUCAO DE E/S

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 13

RELOGIO DO SISTEMA= 263
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 265

ENTROU NO CARREGADOR
VAI CARREGAR O PROG. DO USUARIO 14
ENTROU NO PROCESSO DE E/S USUARIO

RELOGIO DO SISTEMA= 265
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 13
RELOGIO DO SISTEMA= 267
CAUSA DA SUSPENSAC FCI INSTRUCAO DE E/S

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 14

RELOGIO DO SISTEMA= 267
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 269

ENTROU NO PROCESSO DE E/S USUARIO

ENTROU NO PROCESSO DE E/S USUARIO

RELOGIO DO SISTEMA= 269
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 271

IMPRESSAO DOS RESULTADOS

PROGRAMA DO USUARIO 7 TENTATIVA DE LER/ESCREVER APÓS TER ACABADO DADOS DA PAGINA DE IMPRESSAO

RELOGIO DO SISTEMA= 271
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 273
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 273
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 275
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 4

RELOGIO DO SISTEMA= 275
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 4
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 4
RELOGIO DO SISTEMA= 277
CAUSA DA SUSPENSÃO FOI FIM OU ERRO

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO POOL DE SAIDA
POOL DE SAIDA DO PROG. DO USUARIO 4
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA= 277
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA= 279
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 5

RELOGIO DO SISTEMA= 279
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 5
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 281

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 281

VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 5

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 5
RELOGIO DO SISTEMA= 2E3
CAUSA DA SUSPENSAC FOI FIM OU ERRO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 2

RELOGIO DO SISTEMA= 2E3
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 2
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DE PERIFERICO
RELOGIO DO SISTEMA= 2E5

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 5

RELOGIO DO SISTEMA= 2E5
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 2

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 2E7
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA= 2E7
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DE PERIFERICO
RELOGIO DO SISTEMA= 2E9

RELOGIO DO SISTEMA= 2E9
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DE PERIFERICO
RELOGIO DO SISTEMA= 2E1

IMPRESSAO DO PROGRAMA DO USUARIO 4
CLC,PE
CLT,RE
LDR,IM,2
TRCO
SLR,IM,2
14237
IMP,ES,2
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 2E1
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DE PERIFERICO
RELOGIO DO SISTEMA= 2E3

RELOGIO DO SISTEMA= 2E3
VAI COMECAR A EXECUCAO DE PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 2E5
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA= 295
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA= 297
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

RELOGIO DO SISTEMA= 297
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 299
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

RELOGIO DO SISTEMA= 299
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 301

IMPRESSAO DOS RESULTADOS
3762

PROGRAMA DO USUARIO 4 FIM NORMAL

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 301
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 303
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 10

RELOGIO DO SISTEMA= 303
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 10

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 10
RELOGIO DO SISTEMA= 305
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 11

RELOGIO DO SISTEMA= 305
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 11

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 11
RELOGIO DO SISTEMA= 307
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO



ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 18

RELOGIO DO SISTEMA= 307
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 18

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 18
RELOGIO DO SISTEMA= 309
CAUSA DA SUSPENSAO FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 12

RELOGIO DO SISTEMA= 309
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA PODANDO P/ TRATAR INTERRUPTO DO PERIFERICO
RELOGIO DO SISTEMA= 311

IMPRESSAO DO PROGRAMA DO USUARIO 5

CLC,RE
SEI,RE
LCC,IM,A
22767
ACC,IM,A
12000
IMP,ES,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 311
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 12
RELOGIO DO SISTEMA= 313
CAUSA DA SUSPENSAO FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 13

RELOGIO DO SISTEMA= 313
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 13
RELOGIO DO SISTEMA= 315
CAUSA DA SUSPENSAO FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 14

RELOGIO DO SISTEMA= 315
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 317
CAUSA DA SUSPENSAO FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 317
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 319
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERRUPTAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA= 319
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DE PERIFERICO
RELOGIO DO SISTEMA= 321

IMPRESSAC DOS RESULTADOS
44767

PROGRAMA DO USUARIO 5 FIM NORMAL

RELOGIO DO SISTEMA= 321
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA= 323
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 2

RELOGIO DO SISTEMA= 323
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 363
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA= 363
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 378
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA= 378
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA= 380
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

RELOGIO DO SISTEMA= 380
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 382
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO COMPLEMENTA INTERUPÇÃO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

RELOGIO DO SISTEMA= 382
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 9

ENTROU NO COMPLEMENTA INTERUPÇÃO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 423
CAUSA DA SUSPENSÃO FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 10

RELOGIO DO SISTEMA= 423
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 10

ENTROU NO COMPLEMENTA INTERUPÇÃO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 10
RELOGIO DO SISTEMA= 426
CAUSA DA SUSPENSÃO FOI FIM OU ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 10
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 11

RELOGIO DO SISTEMA= 426
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 11

ENTROU NO COMPLEMENTA INTERUPÇÃO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 11
RELOGIO DO SISTEMA= 428
CAUSA DA SUSPENSÃO FOI FIM OU ERRO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 18

RELOGIO DO SISTEMA= 428
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 18
SUSPENDE PROGRAMA PEDANDO P/ TRATAR INTERUPÇÃO DO PERIFERICO
RELOGIO DO SISTEMA= 430

ENTROU NO PROCESSO DE IMPRESSÃO

RELOGIO DO SISTEMA= 430
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 18

ENTROU NO COMPLEMENTA INTERUPÇÃO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 18
RELOGIO DO SISTEMA= 432
CAUSA DA SUSPENSÃO FOI INSTRUÇÃO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 12

RELOGIO DO SISTEMA= 432
VAI COMEÇAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA PEDANDO P/ TRATAR INTERUPÇÃO DO PERIFERICO
RELOGIO DO SISTEMA= 435

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 11

RELOGIO DO SISTEMA= 43E
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 43F

RELOGIO DO SISTEMA= 43F
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 441

IMPRESSAO DO PROGRAMA DO USUARIO 10

LFF,ES,A
IMF,ES,A
STC,FI,E
??
LFP,ES,E
AND,FI,E
??
TMF,ES,E
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 441
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 444

RELOGIO DO SISTEMA= 444
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 451

IMPRESSAO DOS RESULTADOS
??

PROGRAMA DO USUARIO 10 PROTECAO MEMORIA

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 451
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 12
RELOGIO DO SISTEMA= 453
CAUSA DA SUSPENSAO FCI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 13

RELOGIO DO SISTEMA= 453
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 13
RELOGIO DO SISTEMA= 460
CAUSA DA SUSPENSAO FCI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 14

RELOGIO DO SISTEMA= 460
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14

SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 463

IMPRESSAO DO PROGRAMA DO USUARIO 11

LER,ES,A
IMP,ES,A
***,DI,A
11
LSP,CI
11
LCC,CJ,A
11
IMP,ES,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 463
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 470
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

ENTROU NO COMPLEMENTA INTERRUPTAO

ENTROU NO PROCESSO DE F/S USUARIO

ENTROU NO ESCALADOR

ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 470
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 472
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

ENTROU NO COMPLEMENTA INTERRUPTAO

ENTROU NO PROCESSO DE F/S USUARIO

ENTROU NO ESCALADOR

ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA= 472
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 474

IMPRESSAO DOS RESULTADOS
1024

PROGRAMA DO USUARIO 11 COD.OP. INVALIDO

RELOGIO DO SISTEMA= 474
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA= 476
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

ENTROU NO COMPLEMENTA INTERRUPTAO

ENTROU NO PROCESSO DE F/S USUARIO

ENTROU NO ESCALADOR

ESCALOU PARA RODAR PROG. DO USUARIO 2

RELOGIO DO SISTEMA= 476
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2

ENTROU NO COMPLEMENTA INTERRUPTAO

RELOGIO DO SISTEMA= 489
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO A

RELOGIO DO SISTEMA= 489
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 502
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 17

RELOGIO DO SISTEMA= 502
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA= 506
CAUSA DA SUSPENSAC FOI FALTA DE PAGINA

ENTROU NA PAGINACAO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 8

RELOGIO DO SISTEMA= 506
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 8

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 508
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 9

RELOGIO DO SISTEMA= 508
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 9
SUSPENDE PROGRAMA PORQUE P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA= 511

ENTROU NO COMPLEMENTA INTERRUPCAO

RELOGIO DO SISTEMA= 511
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 9

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 561
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 18

RELOGIO DO SISTEMA= 551
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 18

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 18
RELOGIO DO SISTEMA= 572
CAUSA DA SUSPENSAC FOI FIM DO ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PRG. DO USUARIO 18
ENTROU NO ESCALADOR

```

*****
RELOGIO DO SISTEMA= 572
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12
*****
ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAC DO PFCGRAMA DO USUARIO 12
RELOGIO DO SISTEMA= 574
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 13
*****
RELOGIO DO SISTEMA= 574
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 574
*****
ENTROU NO PROCESSO DE IMPRESSAC

RELOGIO DO SISTEMA= 574
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13
*****
ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAC DO PROGRAMA DO USUARIO 13
RELOGIO DO SISTEMA= 578
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 14
*****
RELOGIO DO SISTEMA= 578
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 580
*****
RELOGIO DO SISTEMA= 580
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14
*****
ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAC DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 582
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15
*****
RELOGIO DO SISTEMA= 582
VAI COMECAR A EXECUCAC DO PROG. DO USUARIO 15
*****
ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 584
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16
*****
RELOGIO DO SISTEMA= 584
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 586
*****

```

- 901 -

- 107 -

STC,CI,A
20
TFR,PE,E
1
CCR,PE,E
REC,CI
27
STC,CI,E
20
CCR,PE,E
EEC,CI
27
ACC,CI,A
20
CCR,PE,E
REC,CI
21
BPA,CI
14
LCC,CI,E
20
STC,CI,A
20
BPA,CI
6
IMF,ES,A
FIM,FS

```

*****
*****ENTROU NO PROCESSO DE IMPRESSAO*****
RELOGIO DO SISTEMA= 584
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
*****
*****ENTROU NO COMPLETA INTERUPCAO*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA= 588
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
*****ENTROU NO PROCESSO DE F/S USUARIO*****
*****ENTROU NO ESCALADOR*****
*****ESCALOU PARA RODAR PROG. DO USUARIO 2*****
RELOGIO DO SISTEMA= 588
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 2
*****
*****ENTROU NO COMPLETA INTERUPCAO*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 2
RELOGIO DO SISTEMA= 590
CAUSA DA SUSPENSAC FOI FIM,OU ERRO
*****
*****ENTROU NO SPOOL DE SAIDA*****
*****SPOOL DE SAIDA DO PROG. DO USUARIO 2*****
*****ENTROU NO ESCALADOR*****
*****ESCALOU PARA RODAR PROG. DO USUARIO 6*****
RELOGIO DO SISTEMA= 590
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 593
*****
*****RELOGIO DO SISTEMA= 593*****
*****VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6*****
*****SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAO DO PERIFERICO*****
*****RELOGIO DO SISTEMA= 596*****
*****

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 656
CAUSA DA SUSPENSAC FCI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 17

RELOGIO DO SISTEMA= 656
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA= 663
CAUSA DA SUSPENSAC FCI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 9

RELOGIO DO SISTEMA= 663
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 703
CAUSA DA SUSPENSAC FCI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 12

RELOGIO DO SISTEMA= 703
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 12

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 12
RELOGIO DO SISTEMA= 709
CAUSA DA SUSPENSAC FCI FIM DO ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 12
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 13

RELOGIO DO SISTEMA= 709
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 712

ENTROU NO PROCESSO DE IMPRESSAR

RELOGIO DO SISTEMA= 712
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 715

RELOGIO DO SISTEMA= 715
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 13

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 13
RELOGIO DO SISTEMA= 721
CAUSA DA SUSPENSAC FCI FIM DO ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 13
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PRG. DO USUARIO 14

RELOGIO DO SISTEMA= 721
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14

PROGRAMA DO USUARIO 18 TEMPO ESGOTADO

ENTROU NO PROCESSO DE IMPRESSAO

```

*****
RELOGIO DO SISTEMA= 594
VAI COMECAR A EXECUCAC DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 599
*****
RELOGIO DO SISTEMA= 599
VAI COMECAR A EXECUCAC DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 606
*****

```

IMPRESSAO DO PROGRAMA DO USUARIO 2

```

LER,ES,A
IMP,FS,A
LER,FS,B
IMP,FS,E
STC,CI,A
TP
DCP,PE,B
REC,CI
TE
ACF,CI,A
TP
DCP,PF,E
REC,CI
TE
EFA,CI
O
IMP,ES,A
FIM,FS

```

ENTROU NO PROCESSO DE IMPRESSAO

```

*****
RELOGIO DO SISTEMA= 606
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
SUSPENDE A EXECUCAC DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 613
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

```

ENTROU NO COMPLEMENTA INTERRUPCAO

```

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

```

```

*****
RELOGIO DO SISTEMA= 613
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 616
*****

```

IMPRESSAO DOS RESULTADOS

```

10
5
50

```

PROGRAMA DO USUARIO 2 FIM NORMAL

```

*****
RELOGIO DO SISTEMA= 616
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
*****

```

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 724

IMPRESSAO DO PROGRAMA DO USUARIO 12

LFP,ES,A
IMP,ES,A
STC,CI,A
10
LFP,ES,A
IMP,ES,A
ACT,CI,A
10
IMP,ES,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 724
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 727

RELOGIO DO SISTEMA= 727
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 730

RELOGIO DO SISTEMA= 730
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 734

IMPRESSAO DOS RESULTADOS

32727
5210

PROGRAMA DO USUARIO 12 OVERFLOW

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 734
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 14
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 744

IMPRESSAO DO PROGRAMA DO USUARIO 13

LFP,ES,B
IMP,ES,B
STC,CI,B
10
LFP,ES,E
IMP,ES,E
SUP,CI,B
10
IMP,ES,E
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 744
VAI COMECAR A EXECUCAO DO PRG. DO USUARIO 14

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 746

ENTROU NO COMPLEMENTA INTERRUPTAO

CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 746
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA= 754

IMPRESSAC DOS RESULTADOS

10
-32767

PROGRAMA DO USUARIO 13 UNDERFLOW

RELOGIO DO SISTEMA= 754
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 789
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA= 789
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA= 830
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA= 830
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA= 843
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

RELOGIO DO SISTEMA= 843
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA= 883
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA= 883
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA= 885
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

- 111 -

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

RELOGIO DO SISTEMA= 825
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA= 825
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 14

RELOGIO DO SISTEMA= 825
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 14

ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 14
RELOGIO DO SISTEMA= 827
CAUSA DA SUSPENSAC FOI FIM DO ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 14
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA= 827
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 830

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 830
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 832

RELOGIO DO SISTEMA= 833
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 840

IMPRESSAO DO PROGRAMA DO USUARIO 14

LEF,ES,A
IMP,ES,A
STC,CI,A
14
LEF,ES,E
IMP,ES,F
SUE,PE,A
?
ACC,PE,A
1
SUE,FI,A
14
IMP,ES,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA= 840
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA= 851

IMPRESSÃO DOS RESULTADOS

50
12
26

PROGRAMA DO USUARIO 14 FIM NORMAL

```
*****
PELOGIO DO SISTEMA= 95J
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
*****
ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA= 979
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO
*****
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

*****
RELOGIO DO SISTEMA= 979
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16
*****
ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
PELOGIO DO SISTEMA=1007
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

*****
PELOGIO DO SISTEMA=1007
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
PELOGIO DO SISTEMA=1020
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

*****
PELOGIO DO SISTEMA=1020
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
*****
ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
PELOGIO DO SISTEMA=1060
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO
*****
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

*****
PELOGIO DO SISTEMA=1060
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****
ENTROU NO COMPLEMENTA INTERRUPTAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1067
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

*****
PELOGIO DO SISTEMA=1067
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9
*****
```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA=1107
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA=1107
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA=1148
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA=1148
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA=1150
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1150
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1163
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

RELOGIO DO SISTEMA=1163
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA=1203
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA=1203
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1205
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

RELOGIO DO SISTEMA=1205
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9

ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA=1222

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA=1222
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA=1262
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGCTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 16

RELOGIO DO SISTEMA=1262
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 16

ENTROU NO COMPLEMENTA INTERUPCAC

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 16
RELOGIO DO SISTEMA=1264
CAUSA DA SUSPENSAC FOI FIM OU ERRC

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 16
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1264
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA=1267

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA=1267
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA=127C

RELOGIO DO SISTEMA=127C
VAI COMECAR A EXECUCAO DO PPOG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAO DO PERIFERICO
RELOGIO DO SISTEMA=1277

IMPRESSAO DO PROGRAMA DO USUARIO 16

LER,ES,A
TMP,FS,A
LER,FS,E
TMP,FS,E
CLM,FI
27
SUP,PE,A
2
REC,CI
20
RMI,CI
16
TAM,CI
27
RPA,DI
6
AFC,PE,A
2
RPA,CI
27

27
LFC,CI,F
27
IMP,ES,B
IMP,ES,A
FIM,ES

RELOGIO DO SISTEMA=1277
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1284
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

RELOGIO DO SISTEMA=1284
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1287

IMPRESSAC DOS RESULTADOS

1
4
2
2

PROGRAMA DO USUARIO 16 FIM NORMAL

RELOGIO DO SISTEMA=1287
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA=1327
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

RELOGIO DO SISTEMA=1227
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1334
CAUSA DA SUSPENSAC FOI INSTRUCAO DE F/S

RELOGIO DO SISTEMA=1234
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 9

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 9
RELOGIO DO SISTEMA=1336
CAUSA DA SUSPENSAC FOI FIM DO EPRO

ENTROU NO PROCESSO DE IMPRESSAC

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 9

ENTROU NO COMPLEMENTA INTERRUPCAO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 9
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

```
*****
RELOGIO DO SISTEMA=1336
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1336
```

ENTROU NO PROCESSO DE IMPRESSAO

```
*****
RELOGIO DO SISTEMA=1336
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1342
```

```
*****
RELOGIO DO SISTEMA=1342
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1346
```

IMPRESSAO DO PROGRAMA DO USUARIO 9

LFP,FS,A
IMP,FS,A
STC,CI,F
20
TEP,PE,B
1
PCP,PE,P
REC,CI
27
STC,CI,E
20
CCP,PE,E
BFC,CI
27
AFP,CI,A
20
PCP,PE,B
REC,CI
21
EPA,CI
14
LDD,CI,E
20
STC,CI,F
20
EPA,CI
6
IMP,FS,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

```
*****
RELOGIO DO SISTEMA=1346
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1346
```

IMPRESSAO DOS RESULTADOS

720

PROGRAMA DO USUARIO 9 FIM NORMAL

```
*****
RELOGIO DO SISTEMA=1360
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
```

- 117 -

118

```

*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA=1377
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1377
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1390
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8
*****
RELOGIO DO SISTEMA=1390
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA=1405
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17
*****
RELOGIO DO SISTEMA=1405
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1407
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15
*****
RELOGIO DO SISTEMA=1407
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA=1409
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1409
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1422
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S
ENTROU NO COMPLEMENTA INTERRUPTAO
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 8
*****
RELOGIO DO SISTEMA=1422
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 8
*****

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 8
RELOGIO DO SISTEMA=1424
CAUSA DA SUSPENSAC FGI FIM OU ERRO

ENTROU NO COMPLEMENTA INTERUPCAO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO E
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA=1424
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1427

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA=1427
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1430

RELOGIO DO SISTEMA=1430
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERUPCAC DO PERIFERICO
RELOGIO DO SISTEMA=1437

ENTROU NO PROCESSO DE IMPRESSAO

IMPRESSAO DO PROGRAMA DO USUARIO 8

LFF,ES,A
IMP,ES,A
LEF,ES,F
IMF,ES,E
STC,CI,A
1P
DCC,PE,E
BFC,DI
16
ACC,CI,A
1P
CCP,PE,E
BFC,DI
16
EFA,FI
0
IMP,ES,A
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA=1437
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1439
CAUSA DA SUSPENSAC FCI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTA INTERUPCAO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 15

RELOGIO DO SISTEMA=1439
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 15

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 15
RELOGIO DO SISTEMA=1441

ENTROU NO COMPLEMENTA INTERUPCAO

CAUSA DA SUSPENSÃO FOM FIM OU ERRO

ENTROU NO SPOOL DE SAIDA
SPOOL DE SAIDA DO PROG. DO USUARIO 15
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO A

```

*****
RELOGIO DO SISTEMA=1441
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1444
*****
RELOGIO DO SISTEMA=1444
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1447
*****

```

IMPRESSAO DOS RESULTADOS

10
20
200

PROGRAMA DO USUARIO A FIM NORMAL

ENTROU NO PROCESSO DE IMPRESSAO

```

*****
RELOGIO DO SISTEMA=1447
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1450
*****
RELOGIO DO SISTEMA=1450
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DO PERIFERICO
RELOGIO DO SISTEMA=1457
*****

```

IMPRESSAO DO PROGRAMA DO USUARIO 15

LEF,ES,A
 IMC,ES,A
 LER,ES,E
 INF,ES,E
 CLM,ET
 27
 SUP,RE,A
 2
 BEC,DI
 20
 PMI,CI
 16
 IAN,DI
 27
 BPA,CI
 6
 ACC,PE,A
 2
 ERA,CI
 27
 IAN,DI
 27
 LFC,CI,E
 27
 IMP,ES,E
 IMP,ES,A


```

*****
RELOGIO DO SISTEMA=1457
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****

```

```

ENTROU NO PROCESSO DE IMPRESSAO
*****
ENTROU NO COMPLEMENTA INTERUPCAO

```

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1464
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

```

```

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

```

```

*****
RELOGIO DO SISTEMA=1464
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****

```

```

ENTROU NO COMPLEMENTA INTERRUPTAC

```

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1466
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

```

```

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

```

```

*****
RELOGIO DO SISTEMA=1466
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTAC DO PERIFERICO
RELOGIO DO SISTEMA=1468
*****

```

```

IMPRESSAO DOS RESULTADOS
144
12
12

```

```

PROGRAMA DO USUARIO 15 FIM NORMAL

```

```

*****
RELOGIO DO SISTEMA=1468
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****

```

```

ENTROU NO COMPLEMENTA INTERRUPTAC

```

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1481
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

```

```

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

```

```

*****
RELOGIO DO SISTEMA=1481
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****

```

```

ENTROU NO COMPLEMENTA INTERRUPTAC

```

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1521
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

```

```

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

```

```

*****
RELOGIO DO SISTEMA=1521
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****

```

```

ENTROU NO COMPLEMENTA INTERRUPTAC

```

```

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1534
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

```

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA=1534
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1575
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1575
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1588
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA=1588
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1629
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1629
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1642
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17

RELOGIO DO SISTEMA=1642
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1687
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1687
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

ENTROU NO COMPLEMENTA INTERUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1695
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR

```

*****
RELOGIO DO SISTEMA=1495
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1736
CAUSA DA SUSPENSAC FOI TEMPO PARCIAL ESGOTADO

*****
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1736
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1746
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17
*****
RELOGIO DO SISTEMA=1749
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1753
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1753
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1764
CAUSA DA SUSPENSAC FOI INSTRUCAO DE E/S

*****
ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 17
*****
RELOGIO DO SISTEMA=1764
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 17
*****
ENTROU NO COMPLEMENTA INTERRUPCAO

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 17
RELOGIO DO SISTEMA=1766
CAUSA DA SUSPENSAC FOI FIM DO ERRO

*****
ENTROU NO SPool DE SAIDA
SPool DE SAIDA DO PROG. DO USUARIO 17
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1766
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO
RELOGIO DO SISTEMA=1769
*****
ENTROU NO PROCESSO DE IMPRESSAO
*****
RELOGIO DO SISTEMA=1769
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPCAO DE PERIFERICO

```

```

RELOGIO DO SISTEMA=1772
*****
RELOGIO DO SISTEMA=1772
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA=1775
*****
RELOGIO DO SISTEMA=1775
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA=1780
*****

```

IMPRESSAO DO PROGRAMA DO USUARIO 17

```

LEP,ES,A
IMP,ES,A
STC,CI,A
  50
LEP,ES,A
IMP,ES,A
STC,CI,A
  51
LEP,ES,A
IMP,ES,A
STC,CI,A
  52
LEP,ES,A
IMP,ES,A
STC,CI,A
  54
LDD,CI,A
  60
ADD,CI,A
  51
ACC,CI,A
  52
ACC,CI,A
  52
ACC,CI,A
  54
LEP,IM,E
  5

```

- 124 -

```

*****
RELOGIO DO SISTEMA=1780
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
*****
ENTROU NO PROCESSO DE IMPRESSAO
*****
SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1785
CAUSA DA SUSPENSAO FOI INSTRUCAO DE F/S
*****
ENTROU NO PROCESSO DE F/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6
*****
RELOGIO DO SISTEMA=1785
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTOES DO PERIFERICO
RELOGIO DO SISTEMA=1790
*****

```

55
SUB,RE,A
2
BEQ,DI
44
BMI,DI
46
INM,DI
55
BRA,DI
34
INM,DI
55
LDD,DI,E
55
IMP,ES,E
FIM,ES

ENTROU NO PROCESSO DE IMPRESSAO

RELOGIO DO SISTEMA=179)
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6
SUSPENDE PROGRAMA RODANDO P/ TRATAR INTERRUPTO DO PERIFERICO
RELOGIO DO SISTEMA=1803

IMPRESSAO DOS RESULTADOS

5
10
6
20
14
11

PROGRAMA DO USUARIO 17 FIM NORMAL

RELOGIO DO SISTEMA=1803
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1805
CAUSA DA SUSPENSAO FOI INSTRUCAO DE E/S

ENTROU NO COMPLEMENTO INTERRUPTO

ENTROU NO PROCESSO DE E/S USUARIO
ENTROU NO ESCALADOR
ESCALOU PARA RODAR PROG. DO USUARIO 6

RELOGIO DO SISTEMA=1805
VAI COMECAR A EXECUCAO DO PROG. DO USUARIO 6

SUSPENDE A EXECUCAO DO PROGRAMA DO USUARIO 6
RELOGIO DO SISTEMA=1811
CAUSA DA SUSPENSAO FOI FIM OU ERRO

ENTROU NO COMPLEMENTO INTERRUPTO

ENTROU NO SPool DE SAIDA
SPool DE SAIDA DO PROG. DO USUARIO 6
UCP PARADA ESPERANDO INTERRUPTO DO PERIFERICO
ENTROU NO PROCESSO DE IMPRESSAO
UCP PARADA ESPERANDO INTERRUPTO DO PERIFERICO
UCP PARADA ESPERANDO INTERRUPTO DO PERIFERICO

IMPRESSAO DO PROGRAMA DO USUARIO 6

LEF,ES,E
LEP,ES,E
STO,DI,A
1E
DCP,RE,E

REC,DI
15
ACC,DI,A
14
IMP,ES,A
CCP,PE,E
REC,DI
15
EPA,CI
7
FIM,ES

IMPRESSAO DOS RESULTADOS

20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200

PROGRAMA DO USUARIO 6 FIM NORMAL

TERMINO DO SISTEMA

ENTROU NO PROCESSO DE IMPRESSAO
UCP PARA A ESPERANDO INTERUPCAO DO PERIFERICO

PASCAL EXECUTION SUMMARY

STEP	STORAGE SUMMARY IN DECIMAL (HEX) BYTES				TIME IN SECONDS
	PROGRAM	STACK	HEAP	UNUSED	
OMP	87F20(0155E0)	14277(CC37CC)	66368(010340)	52888(00CF98)	103.59
IPASCRJ	62296(00F358)	F712C(C1545C)	0(000000)	71632(011700)	27.55

APÊNDICE

III

" Definicão dos tipos do sistema "

CONST CONTPROCESSOS=9;

PAGDISCONT=1024;

PAGMEMCONT=64;

BCPCONT=20;

DUPLOCONT=2;

BUFFERCONT=16;

TYPE ESDISPOSITIVO=(DISCO,IMPRESSORA,LEITORA,MEMORIA,DISPOSITIVO,
VOVIRTUAL);

ESOPERACAO=(ENTRADA,SAIDA,MOVE,CONTROLE);

ESRESULTADO=(COMPLETO,INTERVENCAO,TRANSMISSAO,FALHA,FIMDE
ARQUIVO,FIMMEIO,INICIOMEIO);

ESPARAM=RECORD

OPERACAO:ESOPERACAO;

STATUS:ESRESULTADO;

ARG: INTEGER

END;

FILADEPROCESSOS=ARRAY(.1..CONTPROCESSOS.) OF QUEUE;

PALAVRA= INTEGER;

PAGINA= ARRAY(.0.31.) OF PALAVRA;

BCP= ARRAY (.1..34.) OF PALAVRA;

CONJBGP= ARRAY (. 1.. BCPCONT.) OF BCP;

INDICESBCP= ARRAY (. 1..BCPCONT.) OF INTEGER;

CONJINDICESBCPS= ARRAY (. (2,3,5,6,7,10).) OF INDICESBCPS;

CONJFILASBCPS= ARRAY(. (2,3,5,6,7,10).) OF QUEUE;

INDICESUNICOS= ARRAY (. (1.4.8).) OF INTEGER;

CONJFILASUNICOS= ARRAY (. (1,4,8).) OF QUEUE;

CONJBUFFER= ARRAY (. 1.. BUFFERCONT .) OF PAGINA;

CONJMARCA= ARRAY (. 1 .. BUFFERCONT .) OF INTEGER;
 INDICESBUF= ARRAY (. 1 .. BUFFERCONT .) OF INTEGER;
 CONJINDICESBUF= ARRAY (. 1..3 .) OF INDICESBUF;
 FILADUPLAPROCESSOS= ARRAY (. 1..2 .) OF QUEUE;
 CONJFILASBUF= ARRAY (. 1..3 .) OF FILADUPLAPROCESSOS;
 INDICESPAGDISCO= ARRAY (. 0 .. 1023 .) OF INTEGER;
 INDICESPAGMEN= ARRAY (. 0.. 63.) OF INTEGER;
 REGISTROOCUPADO= RECORD
 INDPAGMEM: INTEGER;
 INDPAGDISCO: INTEGER;
 INDBCP: INTEGER
 END;
 ARQUIVOOCUPADO=ARRAY (. 0..63 .) OF REGISTROOCUPADO;
 REGISTROEQUIVALENCIA= RECORD
 INDPAGMEM, INDPAGDISC: INTEGER
 END;
 REGISTROEQUIVALENCIA=ARRAY (. 1 ..BCPCONT .) OF
 REGISTROEQUIVALENCIA;
 DESCRITOR= ARRAY (. 1..21 .) OF INTEGER;
 SEQUENCIA= (FIMJOB, INICIOJOB, INICIOPROGRAMA, INICIODADOS, MEIOPRO
 GRAMA, MEIODADOS, FIMJOB);
 CONJFILABCP= ARRAY (. (2,3,5,7,10).) OF FILA;
 CONJFILABUF= ARRAY (. 1..3 .) OF FILA;

```

TYPE FILA= CLASS ( LIMITE: INTEGER );
" IMPLEMENTACAO DAS FILAS "
VAR INICIO,FIM: INTEGER;
VAR ENTRY TAMANHO: INTEGER;

FUNCTION ENTRY CHEGADA: INTEGER;
BEGIN
    CHEGADA:=FIM;
    FIM:=FIM MOD LIMITE+1;
    TAMANHO:=TAMANHO + 1
END;

FUNCTION ENTRY PARTIDA: INTEGER;
BEGIN
    PARTIDA:=INICIO;
    INICIO:=INICIO MOD LIMITE + 1;
    TAMANHO:=TAMANHO + 1
END;

FUNCTION ENTRY VAZIA: BOOLEAN;
BEGIN VAZIA:= ( TAMANHO = 0 ) END;

FUNCTION ENTRY CHEIA: BOOLEAN;
BEGIN CHEIA:= ( TAMANHO = LIMITE ) END;

BEGIN
    INICIO:=1, FIM:=1; TAMANHO:=0
END;

```

TYPE RECURSO = MONITOR;

VAR LIVRE: BOOLEAN;

Q: FILADEPROCESSOS;

PROXIMA: FILA;

" R E C U R S O "

PROCEDURE ENTRY REQUISITA;

BEGIN

IF LIVRE THEN LIVRE:= FALSE

ELSE DELAY (Q(. PROXIMA.CHEGADA .))

END;

PROCEDURE ENTRY LIBERA;

BEGIN

IF PROXIMA.VAZIA THEN LIVRE:= TRUE

ELSE CONTINUE(Q(.PROXIMA.PARTIDA.))

END;

BEGIN

LIVRE:= TRUE;

INIT PROXIMA(CONTPROCESSOS)

END;



```

TYPE OPERADISCO= CLASS;

PROCEDURE TRANSFERE (COMANDO: ESOPERACAO; ENDPAGINA: INTEGER; VAR
        BLOCO: PAGINA);

VAR PARAM: ESPARAM;

BEGIN

    WITH PARAM

    DO BEGIN

        OPERACAO:=COMANDO;

        ARG:=ENDPAGINA;

        IO (BLOCO, PARAM, DISCO);

        WHILE STATUS <> COMPLETO

        DO IO (BLOCO, PARAM, DISCO)

    END

END;

PROCEDURE ENTRY LE (ENDPAGINA: INTEGER; VAR BLOCO: UNIV PAGINA);

BEGIN

    TRANSFERE (ENTRADA, ENDPAGINA, BLOCO)

END;

PROCEDURE ENTRY ESCRIBE (ENDPAGINA: INTEGER; VAR BLOCO: UNIV PAGINA);

BEGIN

    TRANSFERE (SAIDA, ENDPAGINA, BLOCO)

END;

BEGIN                                END;

```

TYPE MOVEDISCO = CLASS (USODISCO: RECURSO);

VAR OPERACAO:OPERADISCO;

PROCEDURE ENTRY LE (ENDPAGINA: INTEGER; VAR BLOCO:PAGINA);

BEGIN

USODODISCO.REQUISITA;

OPERACAO.LE (ENDPAGINA, BLOCO);

USODODISCO.LIBERA

END;

PROCEDURE ENTRY ESCRIVE (ENDPAGINA: INTEGER; VAR BLOCO:PAGINA);

BEGIN

USODODISCO.REQUISITA;

OPERACAO.ESCREVE (ENDPAGINA, BLOCO);

USODODISCO.LIBERA

END;

BEGIN

END;

TYPE OPERAMEMORIA= CLASS;

PROCEDURE TRANSFERE (COMANDO:ESOPERACAO;ENDPAGINA:INTEGER;

VAR BLOCO: PAGINA);

VAR PARAM: ESPARAM;

BEGIN

WITH PARAM

DO BEGIN

OPERACAO:=COMANDO;

ARG:=ENDPAGINA;

IO (BLOCO ,PARAM,MEMORIA) ;

WHILE STATUS <> COMPLETO

DO IO (BLOCO,PARAM,MEMORIA)

END

END;

PROCEDURE ENTRY LE (ENDPAGINA:INTEGER; VAR BLOCO:UNIV PAGINA);

BEGIN

TRANSFERE (ENTRADA,ENDPAGINA,BLOCO);

END;

PROCEDURE ENTRY ESCREVE (ENDPAGINA:INTEGER; VAR BLOCO: UNIV PAGINA);

BEGIN

TRANSFERE (SAIDA,ENDPAGINA,BLOCO)

END;

BEGIN

END;

TYPE MOVEMEMORIA= CLASS (USODAMEMORIA: RECURSO);

VAR OPERACAO:OPERAMEMORIA;

PROCEDURE ENTRY LE (ENDPAGINA:INTEGER;VAR BLOCO;PAGINA);

BEGIN

USODAMEMORIA.REQUISITA;

OPERACAO.LE (ENDPAGINA,BLOCO);

USODAMEMORIA.LIBERA

END;

PROCEDURE ENTRY ESCREVE (ENDPAGINA:INTEGER; VAR BLOCO: PAGINA);

BEGIN

USODAMEMORIA.REQUISITA;

OPERACAO.ESCREVE (ENDPAGINA,BLOCO);

USODAMEMORIA.LIBERA

END;

BEGIN

INIT OPERACAO

END;

TYPE BCPS= MONITOR;

VAR MATRIZBCP: CONJBCP;

PROCEDURE ENTRY LE (INDICEBCP:INTEGER; VAR BLOCO:BCP);

BEGIN

BLOCO:=MATRIZBCP (. INDICEBCP .)

END;

PROCEDURE ENTRY ESCREVE (INDICEBCP:INTEGER;BLOCO:BCP);

BEGIN

MATRIZBCP (. INDICEBCP .) :=BLOCO

END;

BEGIN

END;

```

TYPE CONTROLABCPS= MONITOR;
VAR CONJFILA:CONJFILABCP;CONJFILAS:CONJFILASBCPS;
      CONJINDICES:CONJINDICESBCPS;
      INDICESDEUM:INDICESUNICOS;
      CONJFILASENVIA,CONJFILASRECEBE:CONJFILASUNICOS;
PROCEDURE ENTRY SOLTA (VAR INDICEBCP:INTEGER; FILAORIGEM:
                          INTEGER );
BEGIN
  IF CONJFILA (. FILAORIGEM .).VAZIA
  THEN DELAY( CONJFILAS(.FILAORIGEM.));
  INDICEBCP:= CONJINDICES(.FILAORIGEM.)(.CONJFILA(.FILAORIGEM.).
                                     PARTIDA.)
END;
PROCEDURE ENTRY MOVE(INDICEBCP,FILADESTINO:INTEGER);
BEGIN
  CONJINDICES(.FILADESTINO.) (CONJFILA(.FILADESTINO.).CHEGADA:):
    = INDICEBCP;
  CONTINUE(CONJFILAS(.FILADESTINO.))
END;
PROCEDURE ENTRY SOLTAUNICO(VAR INDICEBCP:INTEGER;FILADEUM: INTE
                              GER);
BEGIN
  IF INDICESDEUM(.FILADEUM.)= -1
  THEN DELAY(CONJFILASRECEBE(. FILADEUM .));
  INDICESDEUM(.FILADEUM.):= -1;
  CONTINUE(CONJFILASENVIA(.FILADEUM.))
END;

```

PROCEDURE ENTRY MOVEUNICO (INDICEBCP, FILADEUM: INTEGER);

BEGIN

IF INDICESDEUM(.FILADEUM.) <> -1
THEN DELAY (CONJFILASENVIA(.FILADEUM.));
INDICESDEUM(.FILADEUM.) := INDICEBCP;
CONTINUE (CONJFILASRECEBE(.FILADEUM.))

END;

PROCEDURE INICIACAO;

VAR I: INTEGER;

BEGIN

INIT CONJFILA(.2.) (BCPCONT), CONJFILA(.3.) (BCPCONT),
CONJFILA(.5.) (BCPCONT), CONJFILA(.6.) (BCPCONT),
CONJFILA(.7.) (BCPCONT), CONJFILA(.10.) (BCPCONT);

FOR I:=1 TO BCPCONT

DO CONJINDICES(.10.) (.CONJFILAS(.10.) .CHEGADA) := I;

INDICESDEUM(.1.) := -1;

INDICESDEUM(.4.) := -1;

INDICESDEUM(.8.) := - 1;

END;

BEGIN INICIACAO END;

TYPE BUFFERES = MONITOR;

VAR MATRIZBUFFER:CONJBUFFER; MARCAS; CONJMARCA;

PROCEDURE ENTRY LE (INDICE:INTEGER; VAR BLOCO:PAGINA; VAR MARCA:
INTEGER);

BEGIN

BLOCO:=MATRIZBUFFER(.INDICE.);

MARCA:=MARCAS(.INDICE.)

END;

PROCEDURE ENTRY ESCREVE (INDICE:INTEGER; BLOCO, PAGINA, MARCA:INTE
GER);

BEGIN

MATRIZBUFFER(.INDICE.):= BLOCO;

MARCAS(.INDICE.):= MARCA

END;

PROCEDURE ENTRY ESCREVEMARCA (INDICE, MARCA:INTEGER);

BEGIN

MARCAS(.INDICE.):= MARCA

END;

BEGIN

END;

```

TYPE CONTROLABUFFERS = MONITOR;
VAR FILAS;CONJFILAS:CONJFILABUF;CONJINDICES:CONJINDICESBUF;
    CONJFILA:CONJFILASBUF;

PROCEDURE ENTRY SOLTA(VAR INDICEBUF:INTEGER; FILAORIGEM:INTEGER);
BEGIN
    IF FILAS(.FILAORIGEM.).VAZIA
    THEN DELAY(CONJFILA(.FILAORIGEM.)(.CONJFILAS(.FILAORIGEM.)(.CHE
        GADA.))));
    INDICEBUF:=CONJINDICE(.FILAORIGEM.)(.FILAS(.FILAORIGEM.)(.PARTI
        DA.)
    END;

PROCEDURE ENTRY MOVE(VAR INDICEBUF:INTEGER; FILADESTINO:INTEGER);
BEGIN
    CONJINDICES(.FILADESTINO.)(.FILAS(.FILADESTINO.).CHEGADA.):=
        INDICEBUF;
    IF (CONJFILAS(.FILADESTINO.)).VAZIA= FALSE
    THEN CONTINUE(CONJFILA(.FILADESTINO.)(.CONJFILAS(.FILADESTINO.)
        .PARTIDA.))
    END;

```

PROCEDURE INICIACAO;

VAR I: INTEGER;

BEGIN

FOR I:=1 TO 3

DO BEGIN

INIT FILAS (.I.) (BUFFERCONT);

INIT CONJFILAS (.I.) (DUPLOCONT)

END;

FOR I:=1 TO BUFFERCONT

DO CONJINDICES (.3.) (.FILAS (.3.) .CHEGADA.) := I-1

END;

BEGIN INICIACAO END;

TYPE CONTROLAPAGDISCO = MONITOR;

VAR FILAPAGLIVRE:FILA; ENCADEAMENTO:INDICESPAGDISCO;

Q:QUEUE; INDICEPAGLIVRE: INDICESPAGDISCO; I:INTEGER;

PROCEDURE ENTRY ALOCAPAGINA (VAR INDICEPAGINA: INTEGER; INDICEAN
TERIOR:INTEGER);

BEGIN

IF FILAPAGLIVRE.VAZIA THEN DELAY (Q);

INDICEPAGINA := INDICEPAGLIVRE (. FILAPAGLIVRE.SAIDA.);

IF INDICEANTERIOR <> -1

THEN ENCADEAMENTO (. INDICEANTERIOR.) := INDICEPAGINA

END;

PROCEDURE ENTRY LIBERAPAGINA (INDICEPAGINA, INDICEPOSTERIOR: INTE
GER);

BEGIN

INDICEPAGLIVRE (. FILAPAGLIVRE.CHEGADA.) := INDICEPAGINA;

INDICEPOSTERIOR := ENCADEAMENTO (. INDICEPAGINA.);

CONTINUE (Q)

END;

PROCEDURE ENTRY ACHAPAGINA (VAR INDICEPAG : INTEGER; PAGFALTOSA,
INDICEINIC : INTEGER);

VAR INDICEINICIAL : INTEGER;

BEGIN

INDICEINICIAL := INDICEINIC;

IF PAGFALTOSA <> 0

THEN FOR I := 1 TO PAGFALTOSA

DO BEGIN

INDICEPOSTERIOR := ENCADEAMENTO (.INDICEINICIAL.);

INDICEINICIAL := INDICEPOSTERIOR

END;

INDICEPAG := INDICEINICIAL

END;

PROCEDURE ENTRY PROXIMAPAGINA (INDICEPAG : INTEGER; VAR INDICE
PROXPAG : INTEGER);

BEGIN

INDICEPROXPAG := ENCADEAMENTO (.INDICEPAG.)

END;

PROCEDURE ENTRY ENCADEIAPAGINA (INDICEPAG, INDICEPROXPAG : INTEGER
);

BEGIN

ENCADEAMENTO (.INDICEPAG.) := INDICEPROXPAG

END;

BEGIN

INIT FILAPAGLIVRE (PAGDISCONT);

FOR I:= 1 TO PAGDISCONT

DO INDICEPAGLIVRE (.FILAPAGLIVRE.CHEGADA.):= I - 1

END;

```

TYPE CONTROLAPAGMEM = MONITOR;
VAR FILAPAGLIVRE, FILAPAGOCUP: FILA; Q: QUEUE;
    INDICEPAGLIVRE, INDICEPAGOCUP: INDICESPAGMEM;
    ARQS: ARQUIVOOCUPADO; INDPAGFIXA, APONTAPAGOCUP: INTEGER;

PROCEDURE ENTRY ALOCAPAGINALIVRE (VAR INDICEPAG: INTEGER) ;
BEGIN
    IF FILAPAGLIVRE.VAZIA
    THEN DELAY (Q) ;
    INDICEPAG := INDICEPAGLIVRE (.FILAPAGLIVRE.SAIDA.)
END;

PROCEDURE ENTRY ATUALIZAFILAOCUPADOS (ARQ: REGISTROOCUPADO) ;
BEGIN
    ARQS (.FILAPAGOCUP.CHEGADA.) := ARQ
END;

PROCEDURE ENTRY FIXAPAGINA (INDICEPAG: INTEGER) ;
BEGIN
    INDPAGFIXA := INDICEPAG
END;

PROCEDURE ENTRY NORMALIZAPAGINA (INDPAGFIXA: INTEGER) ;
BEGIN
    INDPAGFIXA := -1
END;

```

PROCEDURE ENTRY ALOCAQUALQUERPAGINA (VAR ARQ;REGISTROOCUPADO;VAR
SUBST:BOOLEAN);

VAR OK: BOOLEAN;

BEGIN

IF FILAPAGLIVRE.VAZIA = FALSE

THEN BEGIN

ARQ.INDPAGMEM:=INDICEPAGLIVRE (.FILAPAGLIVRE.SAIDA.);

SUBST:=FALSE

END

ELSE BEGIN

REPEAT

APONTAPAGOCUP:=FILAPAGOCUP.SAIDA;

IE ARQS (.APONTAPAGOCUP.).INDPAGMEM = INDPAGFIXA

THEN BEGIN

ARQS (.FILAPAGOCUP.CHEGADA.):= ARQS (.APONTAPAGOCUP.
);

OK:= FALSE

END

ELSE OK:= TRUE

UNTIL OK;

ARQ:= ARQS (.APONTAPAGOCUP.);

SUBST:= TRUE

END

END;

```

PROCEDURE ENTRY LIBERAPAGINA ( INDICEPAG:INTEGER) ;
VAR OK: BOOLEAN;
BEGIN
    REPEAT
        APONTAPAGOCUP:=FILAPAGOCUP.SAIDA;
        IF ARQS(.APONTAPAGOCUP).INDPAGMEM <> INDICEPAG
            THEN BEGIN
                ARQS(.FILAPAGOCUP.CHEGADA).:=ARQS(.APONTAPAGOCUP.);
                OK:= FALSE
            END
        ELSE BEGIN
            INDICEPAGLIVRE(.FILAPAGLIVRE.CHEGADA).:=INDICEPAG;
            OK:= TRUE
        END
    UNTIL OK;
    CONTINUE(Q)
END;
PROCEDURE INICIACAO;
VAR I: INTEGER;
BEGIN
    INIT FILAPAGLIVRE (PAGMEMCONT) ,FILAPAGOCUP (PAGMEMCONT) ;
    FOR I:= 1 TO PAGMEMCONT
        DO INDICEPAGLIVRE(.FILAPAGLIVRE.CHEGADA).:=I - 1
    END;

BEGIN    INICIACAO    END;

```

```

TYPE DESCRITOJOB = MONITOR
VAR BLOCODESCRITOR:DESCRITOR; Q1,Q2: QUEUE;
    PARAM: ESPARAM;
PROCEDURE ENTRY SOLICITASINCRONIZACAO;
BEGIN
    DELAY(Q1)
END;
PROCEDURE ENTRY CONCEDESINCRONIZACAO;
BEGIN
    CONTINUE(Q1)
END;
PROCEDURE ENTRY MOVEIN(VAR BLOCODESCRITOR: DESCRITOR);
BEGIN
    PARAM.OPERACAO:=ENTRADA;
    IO(BLOCODESCRITOR,PARAM,DISPOSITIVOVIRTUAL)
END;
PROCEDURE ENTRY MOVEOUT(BLOCODESCRITOR:DESCRITOR);
BEGIN
    PARAM.OPERACAO:=SAIDA;
    IO(BLOCODESCRITOR,PARAM,DISPOSITIVOVIRTUAL)
END;
PROCEDURE ENTRY ESPERASINAL;
BEGIN
    DELAY(Q2) "O CONTINUE E DADO PELO NUCLEO NO TRATAMENTO DE IN
              TERRUPCAO"
END;

BEGIN                                END;

```

" PROCESSO DE LEITURA:

PEDE UM BUFFER PARA A FILA DE BUFFERS LIVRES;

PEDE UMA OPERACAO DE E/S PARA PREENCHER O BUFFER;

MOVE O BUFFER PARA A FILA DOS PREENCHIDOS PELA LEITURA.

"

```

TYPE LEITURA = PROCESS
(CONTROLABUFFERS:CONTROLABUFFERS;BUFFERSFISICO:BUFFERES);
VAR PARAM: ESPARAM;BLOCOBUF:PAGINA;INDICEBUF:INTEGER;
    OK:BOOLEAN;
BEGIN ----
    PARAM.OPERACAO:= ENTRADA;PARAM.ARG:=0;
CYCLE
REPEAT
    CONTROLBUFFERS.SOLTA(INDICEBUF,3);
    IO(BLOCOBUF,PARAM,LEITORA);
    CASE PARAM.STATUS
    OF COMPLETO: OK:= TRUE;
        INTERVENCAO: BEGIN
            OK:= FALSE;
            CONTROLBUFFERS.MOVE(INDICEBUF,3);
            WAIT
            END;
        TRANSMISSAO,FALHA:BEGIN
            OK:= FALSE;
            WAIT
            END
        END
    UNTIL OK;
    BUFFERSFISICO.ESCREVE(INDICEBUF;BLOCOBUF,0);
    CONTROLBUFFERS.MOVE(INDICEBUF,1) "MOVE BUFFER PARA A FILA
    DOS PREENCHIDOS PELA LEITURA"
END
END;

```

" SPOOL DE ENTRADA:

PEGA UM BUFFER DA FILA DOS PREENCHIDOS PELA LEITURA;ANALISA
O BUFFER;

SE INICIO DE PROGRAMA PEDE UM BCP PARA A FILA DE BCP's LI
VRES;

ANALISA O BUFFER PARA MONTAR A BCP;

CARREGA O PROGRAMA NO DISCO;

MOVE O BCP PARA A FILA DE RESIDENTES(2) OU ACABADOS(7);

CONFORME O RESULTADO DA ANALISE DO BUFFER

MOVE O BUFFER PARA A FILA DE BUFFERS LIVRES"


```

TYPE SPOOLDEENTRADA = PROCESS
(CONTROLBCPS:CONTROLABCPS;BCPSFISICO;CONTROLBUFFERS:CONTROLABUF
FERS;BUFFERSFISICO:BUFFERES;CONTROLPAGDISCO:CONTROLAPAGDISCO ;
USODISCO:RECURSO);
VAR MOVEDISC:MOVEDISCO;BLOCOBUF:PAGINA;SEQ:SEQUENCIA;
      BLOCO:BCPINDICEBUF,INDICEBCP,NUMPAG,INDICEPAG,  INDICEANTE
      RIOR,I,INDICEPAGDISCO: INTEGER;
BEGIN
MOVEDISC(USODISCO);
SEQ:= FIMJOB;
CYCLE
CONTROLBUFFERS.SOLTA(INDICEBUF,1): "PEGA 1 BUFFER DA FILA  DOS
PREENCHIDOS PELA LEITURA"
BUFFERESFISICO.LE(INDICEBUF,BLOCOBUF,MARCA);
IF BLOCOBUF(.1.) = 77777 "CARTAO DE CONTROLE DE JOB"
THEN BEGIN  "(1)"
      IF SEQ <> FIMJOB
      THEN BEGIN  "(2)"
        BLOCOBCP(.34.):=70; "CARTAO DE JOB FORA DE LUGAR"
        BCPFISICO.ESCREVE(INDICEBCP,BLOCOBCP);
        CONTROLBCPS.MOVE(INDICEBCP,7); "MOVE O BCP P/ A FI
        LA DE ACABADOS"
        SEQ:= FIMJOB
      END  " FIM (2)"
      ELSE BEGIN  "(3)"
        CONTROLBCPS.SOLTA(INDICEBCP,10); "PEGA UM  BCP  DA

```

FILA DOS LIVRES"

BLOCOBCP(.1.):=BLOCOBUF(.1.); "IDENTIFICADOR DE JOB"

BLOCOBCP(.8.):=BLOCOBUF(.2.); "TEMPO ESTIMADO DE EXECUCAO"

NUMPAG:=BLOCOBUF(.3.); "Nº DE PAGINAS ESTIMADAS DE IMPRESSAO"

IF NUMPAG > 0

THEN BEGIN "(4)"

INDICEANTERIOR:= -1;

CONTROLPAGDISC.ALOCAPAGINA(INDICEPAGINA, INDICEANTERIOR);

INDICEANTERIOR:=INDICEPAGINA;

BLOCOBCP(.31.):=INDICEPAGINA; "END. INICIAL DA AREA DE IMPRESSAO"

BLOCOBCP(.33.):=INDICEPAGINA; "END. FINAL DA AREA DE IMPRESSAO"

NUMPAG:=NUMPAG -1;

IF NUMPAG > 0

THEN BEGIN "(5)"

FOR I:= 1 TO NUMPAG

DO BEGIN "(6)"

CONTROLPAGDISC.ALOCAPAGINA(INDICEPAGINA, INDICEANTERIOR); INDICEANTERIOR :=INDICEPAGINA

END; "FIM (6)"

BLOCOBCP(.33.):=INDICEPAGINA

END; "FIM (5)"

```

                END                "FIM (4) "
        SEQ:=INICIOJOB
        END                "FIM (3) "
        END                "FIM (1) "
ELSE IF BLOCUBUF(.1.)=77776 "CARTAO CONTROLE DE PROGRAMA"
        THEN BEGIN
                IF SEQ <> INICIOJOB
                THEN BEGIN
                        BLOCBCP(.34.):=60;"CARTAO DE PROGRAMA
                        FORA DE LUGAR"
                        BCPSFISICO.ESCREVE(INDICEBCP,BLOCBCP);
                        CONTROLBCPS.MOVE(INDICEBCP,7);
                        SEQ:= FIMJOB
                END
                ELSE SEQ:=INICIOPROGRAMA
        END
ELSE IF BLOCUBUF(.1.)=77775 "CARTAO CONTROLE DADOS"
        THEN BEGIN
                IF SEQ <> MEIOPROGRAMA
                THEN BEGIN
                        BLOCBCP(.34.):=50;"CARTAO DE DADOS
                        FORA DO LUGAR"
                        BCPFISICO.ESCREVE(INDICEBCP, BLOCO
                        BCP);
                        CONTROLBCPS.MOVE(INDICEBCP,7);
                        SEQ:= FIMJOB
                END

```

```

        ELSE SEQ:=INICIODADOS
    END
ELSE IF BLOCOBUF(.1.)=77774 "CARTAO FIM JOB"
    THEN BEGIN
        IF (SEQ=MEIOPROGRAMA) OR (SEQ=MEIODA
        DOS)
        THEN BEGIN
            BCPSFISICO.ESCREVE(INDICEBCP ,
            BLOCOBCP);
            CONTROLBCPS.MOVE(INDICEBCP,2);
            SEQ:=FIMJOB
        END
    ELSE IF SEQ <> FIMJOB
    THEN BEGIN
        BLOCOBCP(.34.):=40;" CAR
        TAO FIM JOB FORA LUGAR"
        BCPSFISICO.ESCREVE(INDICE
        BCP,BLOCOBCP);
        CONTROLBCPS.MOVE( INDICE
        BCP, 7);
        SEQ:=FIMJOB
    END
    END
ELSE IF (SEQ=INICIOPROGRAMA) OR (SEQ=
    INICIODADOS)
    THEN BEGIN

```

```

CONTROLPAGDISCO.ALOCAPAGI
NA (INDICEPAGDISC,-1);
INDICEANTERIOR:=INDICEPAG
DISC;
IF SEQ=INICIOPROGRAMA
THEN BEGIN
        BLOCOBCP(.26.):=INDI
        CEPAGDISC;
        "END.INICIAL DO PRO
        GRAMA NO DISCO"
        BLOCOBCP(.27.):=1; "
        TAMANHO DO PROG EM
        PAGINAS"
        BLOCOBCP(.28.):=1; "
        END.INICIAL DA AREA
        DE DADOS"
        SEQ:=MEIOPROGRAMA
END
ELSE BEGIN
        BLOCOBCP(.28.):=INDI
        CEPAGDISC;
        BLOCOBCP(.30.):=INDI
        CEPAGDISC;
        SEQ:=MEIODADOS
END;
MOVEDISC.ESCREVE (INDICEPAG
                                DISC; BLO
                                COBUF);

```

END

ELSE SEQ:=FIMJOB;

CONTROLBUFFERS.MOVE (INDICEBUF,3)

END

END;

" CARREGADOR:

PEDE UM BCP DA FILA DE RESIDENTES (2);

PEDE PARA A FILA DE PAGINAS DE MEMORIA LIVRE UMA PAGINA PA
RA MONTAR A TABELA DE PAGINAS;

PEDE NOVA PAGINA PARA CARREGAR A PRIMEIRA PAGINA DO PROGRA
MA;

MOVE O BCP PARA A FILA DE PRONTOS (3)

"

```

TYPE CARREGADOR = PROCESS
(CONTROLBCPS:CONTROLABCPS;BCPSFISICO:BCPS;USODODISCO:RECURSO ;
CONTROLPAGMEM:CONTROLAPAGMEM;EQUIVALENCIAMD:EQUIVALENCIAMEM
DISC;USUDAMEMORIA:RECURSO);
VAR MOVEMEM:MOVEMEMORIA;MOVEDISC:MOVEDISCO:BLOCOBCP:BCP; BLOCO
PAG:PAGINA;
REGISTROEQUIVALENCIA;INDICEBCP,INDICETABPAG,I,INDICEPAGMEM,
INDICEPAGDISC:INTEGER;
BEGIN
INIT MOVEMEM(USODAMEMORIA),MOVEDISC(USODODISCO);
CYCLE
CONTROLBCPS.SOLTA(INDICEBCP,2);
BCPFISICO.LE(INDICEBCP,BLOCOBCP);
CONTROLPAGMEM.ALOCAPAGINALIVRE(INDICETABPAG);
BLOCOBCP(.2.):=INDICETABPAG;
FOR I:=0 TO 31 DO BLOCOPAG(.I.):= -16384;
MOVEMEM.ESCREVE(INDICETABPAG,BLOCOPAG);
CONTROLPAGMEM.ALOCAPAGINALIVRE(INDICEPAGMEM);
INDICEPAGDISC:=BLOCOBCP(.26.);
MOVEDISC.LE(INDICEPAGDISC,BLOCOPAG);
MOVEMEM.ESCREVE(INDICEPAGMEM,BLOCOPAG);
MOVEMEM.LE(INDICETABPAG,BLOCOPAG);
BLOCOPAG(.0.):=INDICEPAGMEM;
MOVEMEM.ESCREVE(INDICETABPAG,BLOCOPAG);
REGISTROEQUIV.INDPAGMEM:=INDICEPAGMEM;
REGISTROEQUIV.INDPAGDISC:=INDICEPAGDISC;
EQUIVALENCIAMD.ENVIA(INDICEBCP,REGISTROEQUIV);

```


BLOCOBCP(.34.):= 1; "INDICA QUE VEIO DO CARREGADOR"

BCPSFISICO.ESCREVE(INDICEBCP,BLOCOBCP);

CONTROLBCPS.MOVE(INDICEBCP,3)

END

END;

" ESCALADOR:

PEDE UM BCP DA FILA DOS PRONTOS(3);

MONTA O BLOCO DESCRITOR, OU SEJA, CRIA O PROCESSO A NIVEL
DO NUCLEO;

MOVE O BCP PARA A FILA DO ESCALADO PARA RODAR(4).

"

TYPE ESCALADOR = PROCESS

(CONTROLBCPS:CONTROLABCPS;BCPSFISICO:BCPS;DESCRITORJOBUS:DESCRITORJOB;EQUIVALENCIA:EQUIVALENCIAMEMDISC;CONTROLPAGMEM:CONTROLAPAGMEM);

VAR BLOCOBCP:BCP;BLOCODESCRITOR:DESCRITOR;REGISTROEQUIV: REGISTROEQUIVALENCIA;REGISTROOCUP:REGISTROOCUPADO;I,INDICEBCP,INDICEPAGMEM:INTEGER;

BEGIN

CYCLE

DESCRITORJOBUS.SOLICITASINCRONIZACAO;

CONTROLBCPS.SOLTA(INDICEBCP,3);

BCPSFISICO.LE(INDICEBCP,BLOCOBCP);

FOR I:=1 TO 8 DO BLOCODESCRITOR(.I.):=BLOCOBCP(.I+1.);

FOR I:=10 TO 21 DO BLOCODESCRITOR(.I.):=BLOCOBCP(.I+1.);

DESCRITORJOBUS.MOVEOUT(BLOCODESCRITOR);

IF (BLOCOBCP(.34.)=0) OR (BLOCOBCP(.34.)=1)

THEN BEGIN

EQUIVALENCIA.RECEBE(INDICEBCP,REGISTROEQUIV);

REGISTROOCUP.INDPAGMEM:=REGISTROEQUIV.INDPAGMEM;

INDPAGMEM:=REGISTROEQUIV.INDPAGMEM;

REGISTROOCUP.INDPAGDISC:=REGISTROEQUIV.INDPAGDISC;

REGISTROOCUP.INDBCP:=INDICEBCP;

CONTROLPAGMEM.ATUALIZAFILAOCUPADOS(REGISTROOCUP);

END

ELSE CONTROLPAGMEM.NORMALIZAPAGINA(INDICEPAGINA);

CONTROLBCPS.MOVEUNICO(INDICEBCP,4)

END

END;

" COMPLEMENTA INTERRUPCAO:

PEDE BCP DA FILA DE ESCALADO PARA RODAR(4);

ATUALIZA BCP COM O DESCRITOR DO PROCESSO;

ANALISA A CAUSA DA INTERRUPCAO, TOMA AS ACOES NECESSARIAS, E

MOVE O BCP PARA A FILA ADEQUADA.

"

```

TYPE COMPLEMENTAINTERRUPCAO = PROCESS
(DESCRITORJOBUS:DESCRITORJOB;CONTROLBCPS:CONTROLABCPS;BCPSFISI
CO:BCPS;USODAMEMORIA:RECURSO;CONTROLPAGMEM:CONTROLAPAGMEM);
VAR MOVEMEM:MOVEMEMORIA;BLOCODESCRITOR:DESCRITOR;
      BLOCOBCP:BCP;BLOCOPAG:PAGINA;
      INDICEBCP,INDICEPAG,INDICETABPAG,I: INTEGER;
BEGIN
  INIT MOVEMEM(USODAMEMORIA);
  CYCLE
    DESCRITORJOBUS.CONCEDESINCRONIZACAO;
    DESCRITORJOBUS.ESPERASINAL;
    CONTROLBCPS.SOLTAUNICO(INDICEBCP,4);
    BCPSFISICO.LE(INDICEBCP,BLOCOBCP);
    DESCRITORJOBUS.MOVEIN(BLOCODESCRITOR);
    FOR I:=2 TO 23 DO BLOCOBCP(.I.):=BLOCODESCRITOR(.I-1.);
    BLOCOBCP(.34.):=BLOCODESCRITOR(.9.);
    BCPSFISICO.ESCRIBE(INDICEBCP,BLOCOBCP);
    CASE BLOCODESCRITOR(.9.)
    OF 1:CONTROLBCPS.MOVE(INDICEBCP,6); "PAGINACAO"
        2,3:CONTROLBCPS.MOVE(INDICEBCP,5); "INSTRUCAO DE E/S"
        4,5,6,7,8,9,11,12:BEGIN
            INDICETABPAG:=BLOCOBCP(.2.);
            MOVEMEM.LE(INDICETABPAG,BLOCOPAG);
            FOR I:=0 TO 31 DO
            IF BLOCOPAG(.I.) <> -16384
            THEN BEGIN

```

INDICEPAG:=BLOCOPAG(.I.);

CONTROLPAGMEM.LIBERAPAGINA(INDICE
PAG)

END;

CONTROLPAGMEM.LIBERAPAGINA(INDICETABPAG);

CONTROLBCPS.MOVE(INDICEBCP,7)

END;

10:CONTROLBCPS.MOVE(INDICEBCP,3) "TEMPO PARCIAL ESGOTADO"

END

END

END;

" PAGINACAO:

PEDE UM BCP PARA A FILA DE PROCESSOS SUSPENSOS POR FALTA DE
PAGINA(6);

VERIFICA SE EXISTE PAGINA DE MEMORIA LIVRE;

SE EXISTIR, COPIA PAGINA FALTOSA NA MESMA,CASO CONTRARIO ,

PROCURA PAGINA PARA SER SUBSTITUIDA;

ATUALIZA A TABELA DE PAGINAS;

MOVE BCP PARA A FILA DE PRONTOS(3).

"

TYPE PAGINACAO = PROCESS

(CONTROLBCPS:CONTROLABCPS;BCPSFISICO:BCPS;CONTROLPAGMEM: CONTROLAPAGMEM;CONTROLPAGDISC:CONTROLAPAGDISCO;USODODISCO,USODAMEMORIA:RECURSO;EQUIVALENCIAMD:EQUIVALENCIAMEMDISC);

VAR MOVEDISC:MOVEDISCO;MOVEMEM:MOVEMEMORIA;BLOCOBCP:BCP; REGISTROOCUP:REGISTROOCUPADO;INDICEBCP,INDICEPAGMEM, INDICEPAGFALT,INDICEPAGINIC,INDICEPAGDISC,INDICEBCPFALT, INDICEPAGDISCDEV;INDICETABPAGFALT,INDICETABPAG,N:INTEGER;

BEGIN

INIT MOVEDISC(USODODISCO),MOVEMEM(USODAMEMORIA);

CYCLE

CONTROLBCPS.SOLTA(INDICEBCP,6);

BCPSFISICO.LE(INDICEBCP,BLOCOBCP);

CONTROLPAGMEM.ALOCAQUALQUERPAGINA(REGISTROOCUP,SUBST);

INDICEPAGMEM:=REGISTROOCUP.INDPAGMEM;

INDICEPAGFALT:=BLOCOBCP(.9.); "PAGINA FALTOSA"

INDICEPAGINIC:=BLOCOBCP(.26.);

CONTROLPAGDISC.ACHAPAGINA(INDICEPAGDISC,INDICEPAGFALT, INDICEPAGINIC);

IF SUBST = FALSE

THEN BEGIN

MOVEDISC.LE(INDICEPAGDISC,BLOCOPAG);

MOVEMEM.ESCREVE(INDICEPAGMEM,BLOCOPAG);

REGISTROEQUIV.INDPAGMEM:=INDICEPAGMEM;

REGISTROEQUIV.INDPAGDISC:=INDICEPAGDISC

END

ELSE BEGIN

INDICEPAGDISCDEV:=REGISTROOCUP.INDPAGDISC;

INDICEBCPALT:=REGISTROOCUP.INDBCP;

BCPSFISICO.LE (INDICEBCPALT, BLOCOBCPALT);

INDICETABPAGALT:=BLOCOBCPALT (.2.);

MOVEMEM.LE (INDICETABPAGALT, BLOCOPAG);

N:=0;

REPEAT

IF BLOCOPAG (.N.)=INDICEPAGMEM

THEN BEGIN

BLOCOPAG (.N.):= -16384;"PALAVRA= 100...0"

OK:=TRUE

END

ELSE BEGIN

N:=N + 1;

OK:=FALSE

END

UNTIL OK;

MOVEMEM.ESCREVE (INDICETABPAGALT, BLOCOPAG);

MOVEMEM.LE (INDICEPAGMEM, BLOCOPAG);

MOVEDISC.ESCREVE (INDICEPAGDISCDEV, BLOCOPAG);

MOVEDISC.LE (INDICEPAGDISC, BLOCOPAG);

MOVEDISC.ESCREVE (INDICEPAGMEM, BLOCOPAG)

END;

INDICETABPAGALT:=BLOCOBCP (.2.);

MOVEMEM.LE (INDICETABPAG, BLOCOPAG);

BLOCOPAG (.INDICEPAGFALT.):=INDICEPAGMEM;

MOVEMEM.ESCREVE (INDICETABPAG , BLOCOPAG) ;
REGISTROEQUIV.INDPAGMEM:=INDICEPAGMEM;
REGISTROEQUIV.INDPAGDISC:=INDICEPAGDISC;
EQUIVALENCIAMD.ENVIA (INDICEBCP , REGSITROEQUIV) ;
BLOCOBCP (.34.) :=0; "INDICA QUE VEIO DO PROCESSO PAGINACAO"
BCPSFISICO.ESCREVE (INDICEBCP , BLOCOBCP) ;
CONTROLBCPS.MOVE (INDICEBCP , 3)

END

END;

" ESUSUARIO:

PEDE UM BCP DA FILA DOS PROCESSOS SUSPENSOS POR PEDIDO DE
E/S (5);

EXECUTA A OPERACAO SOLICITADA;

MOVE O BCP PARA A FILA DOS PRONTOS (3).

"

TYPE ESUSUARIO = PROCESS

(CONTROLBCPS:CONTROLABCPS;BCPSFISICO:BCPS;CONTROLPAGMEM: CONTROL
LAPAGMEM,CONTROLPAGDISC:CONTROLAPAGDISCO;USODAMEMORIA,USODODIS
CO:RECURSO);

VAR MOVEDISC:MOVEDISCO;MOVEMEM:MOVEMEMORIA;BLOCOBCP:BCP; BLOCO
PAG:PAGINA;INDICEBCP,INDICEPAGMEM,APONTADORDADOS,INDICEPROX
PAG,APONTADORIMPRESSAO,CODIGOOPERACAO:INTEGER;

BEGIN

INIT MOVEDISCO (USODODISCO),MOVEMEM (USODAMEMORIA);

CYCLE

CONTROLBCPS.SOLTA (INDICEBCP,5);

BCPSFISOCO.LE (INDICEBCP,BLOCOBCP);

INDICEPAGMEM:=BLOCOBCP (.9.);

CONTROLPAGMEM.FIXAPAGINA (INDICEPAGMEM);

IF CODIGOOPERACAO = 0 "LEITURA"

THEN BEGIN

APONTADORDADOS:=BLOCOBCP (.29.);"APONTADOR CORRENTE DA
AREA DE IMPRESSAO"

MOVEDISC.LE (APONTADORDADOS,BLOCOPAG);

MOVEMEM.ESCREVE (INDICEPAGMEM,BLOCOPAG);

CONTROLPAGDISC.PROXIMAPAGINA (APONTADORDADOS,INDICEPROX
PAG);

BLOCOBCP (.29.):=INDICEPROXPAG

END

ELSE BEGIN "IMPRESSAO"

APONTADORIMPRESSAO:=BLOCOBCP (.32.);"APONTADOR CORRENTE
DA AREA DE IMPRESSAO"

MOVEMEM.LE (INDICEPAGMEM, BLOCOPAG) ;

MOVEDISC.ESCREVE (APONTADORIMPRESSAO, BLOCOPAG) ;

CONTROLPAGDISC.PROXIMAPAGINA (APONTADORIMPRESSAO, INDICE
PROXPAG) ;

BLOCOBCP (.32.) :=INDICEPROXPAG;

END

BLOCOBCP (.34.) :=2; "INDICA QUE VEIO DA ESUSUARIO"

BCPSFISICO.ESCREVE (INDICEBCP, BLOCOBCP) ;

CONTROLBCPS.MOVE (INDICEBCP, 3)

END

END;



" SPOOL DE SAIDA:

PEDE UM BCP DA FILA DE ACABADOS (7);

PEDE UM BUFER PARA A FILA DE BUFFERS LIVRES;

MONTA O BUFFER COM PAGINA DO DISCO E DEVOLVE-A PARA A FILA
DE PAGINAS DO DISCO LIVRES;

MOVE O BUFFER PARA A FILA DOS PREENCHIDOS PELO SPOOL DE
SAIDA (2);

MOVE O BCP PARA A FILA DE PROCESSOS EM FASE DE SAIDA (8)

"

```

TYPE SPOOLDESAIDA = PROCESS
(CONTROLBCPS:CONTROLABCPS;BCPSFISICO:BCPS;CONTROLBUFFERS:CONTRO
LABUFFERS;BUFFERSFISICO:BUFFERS;CONTROLPAGDISC:CONTROLAPAGDIS
CO;USODODISCO:RECURSO);
VAR MOVEDISC:MOVEDISCO;BLOCOBCP:BCP;BLOCOBUF:PAGINA;INDICEBCP ,
      PONTEIROSP,N,I,INDICEBUF,INDICEPROXPAG:INTEGER;
PROCEDURE OPERASP;
BEGIN
CONTROLBUFFERS.SOLTA(INDICEBUF,3);
MOVEDISC.LE(PONTEIROSP,BLOCOBUF);
CONTROLPAGDISC.LIBERAPAGINA(PONTEIROSP,INDICEPROXPAG);BUFFERES
FISICO.ESCREVE(INDICEBUF,BLOCOBUF,0);
CONTROLBUFFERS.MOVE(INDICEBUF,2)
END;
BEGIN
INIT MOVEDISC(USODODISCO);
CYCLE
CONTROLBCPS.SOLTA(INDICEBCP,7);
BCPSFISICO.LE(INDICEBCP,BLOCOBCP);
CONTROLBCPS.MOVEUNICO(INDICEBCP,8);
PONTEIROSP:=BLOCOBCP(.26.);
N:=BLOCOBCP(.27.);
IF N > 0
THEN BEGIN
      FOR I:= 1 TO N
      DO BEGIN

```

```

OPERASP;

PONTEIROSP:=INDICEPROXPAG

END;

IF BLOCOBCP(.28.) <> -1

THEN BEGIN

    INDICEPROXPAG:=BLOCOBCP(.28.); "END. AREA DADOS"

    REPEAT

        PONTEIROSP:=INDICEPROXPAG;

        OPERASP

        UNTIL PONTEIROSP=BLOCOBCP(.30.) "END.FINAL
        AREA DADOS"

    END;

    INDICEPROXPAG:=BLOCOBCP(.31.); "END. INICIAL AREA DE
    IMPRESSAO"

    REPEAT

        PONTEIROSP:=INDICEPROXPAG;

        OPERASP

        UNTIL PONTEIROSP=BLOCOBCP(.33.); "END.FINAL AREA DE
        IMPRESSAO"

        BUFFERESFISICO.ESCREVEMARCA(INDICEBUF,1)

    END

ELSE BEGIN

    FOR I:=0 TO 31 DO BLOCOBUF(.I.):0;

    CONTROLBUFFERS.SOLTA(INDICEBUF,3);

    BUFFERESFISICO.MOVE(INDICEBUF,BLOCOBUF,1);

    CONTROLBUFFERS.MOVE(INDICEBUF,2)

    END

END;

```


" IMPRESSAO:

PEDE O BCP DA FILA DE PROCESSO EM FASE DE SAIDA(8);

PEDE UM BUFFER DA FILA DE BUFFERS PREENCHIDOS PELO SPOOL DE
SAIDA(2);

PEDE OPERACAO DE SAIDA PARA ESCREVER O CONTEUDO DO BUFFER;

MOVE O BUFFER PARA A FILA DOS BUFFERS LIVRES

"

```

TYPE IMPRESSAO = PROCESS
(CONTROLBCPS:CONTROLABCPS,BCPSFISICO:BCPS;CONTROLBUFFERS:CONTRO
LABUFFERS,BUFFERESFISICO:BUFFERES);
VAR BLOCOBCP:BCP;BLOCOBUF:PAGINA;TEXTO:BLOCO;CODCONDICAO,INDICE
BCP,INDICEBUF,I,MARCA:INTEGER;
PROCEDURE SAIDA(BLOCO:UNIV PAGINA);
VAR PARAM:ESPARAM;
BEGIN
PARAM.OPERACAO:=SAIDA;
PARAM.ARG:=0;
IO(BLOCO,PARAM,IMPRESSORA);
IF PARAM.STATUS <> COMPLETO
THEN BEGIN
REPEAT
WAIT;
IO(BLOCO,PARAM,IMPRESSORA)
UNTIL PARAM.STATUS=COMPLETO
END
END;
BEGIN
CYCLE
CONTROLBCPS.SOLTAUNICO(INDICEBCP,8);
BCPSFISICO.LE(INDICEBCP,BLOCOBCP);
REPEAT
CONTROLBUFFERS.SOLTA(INDICEBUF,2);
BUFFERESFISICO.LE(INDICEBUF,BLOCOBUF,MARCA);
IF MARCA=0

```

```

THEN BEGIN
    SAIDA (BLOCOBUF) ;
    CONTROLBUFFERS.MOVE (INDICEBUF, 3)
END
ELSE BEGIN
    SAIDA (BLOCOBUF) ;
    BUFFERESFISICO.ESCREVEMARCA (INDICEBUF, 0) ;
    CONTROLBUFFERS.MOVE (INDICEBUF, 3) ;
    CODCONDICAO:=BLOCOBCP (.34.) ;
    CONTROLBCPS.MOVE (INDICEBCP, 10) ;
    CASE CODCONDICAO
    OF 4:TEXTO:='FIM NORMAL' ;
        5:TEXTO:='PROTECAO DE MEMORIA' ;
        6:TEXTO:='CODIGO DE OPERACAO INVALIDO' ;
        7:TEXTO:='OVERFLOW' ;
        8:TEXTO:='UNDERFLOW' ;
        9:TEXTO:='TEMPO TOTAL ESGOTADO' ;
        11:TEXTO:='UNDERFLOW NA PILHA' ;
        12:TEXTO:='OVERFLOW NA PILHA' ;
        40:TEXTO:='CARTAO DE JOB FORA DO LUGAR' ;
        50:TEXTO:='CARTAO DE DADO FORA DE LUGAR' ;
        60:TEXTO:='CARTAO DE PROGRAMA FORA DO LUGAR' ;
        70:TEXTO:='CARTAO DE JOB FORA DO LUGAR'
    END;
    SAIDA (TEXTO)
    END
UNTIL MARCA=1
END;

```

" PROCESSO INICIAL:

COLOCA TODOS OS PROCESSOS DO SISTEMA EM CONDICOES DE INICI
AR SEU TRABALHO

"

" PROCESSO INICIAL "

VAR USODODISCO,USODAMEMORIA:RECURSO;BCPSFISICO:BCPS;CONTROLBCPS
:CONTROLABCPS;BUFFERRESFISICO:BUFFERES;CONTROLBUFFERS: CON
TROLABUFFERS;CONTROLPAGDISC:CONTROLAPAGDISCO;CONTROLPAGMEM:
CONTROLAPAGMEM;
EQUIVALENCIAMD:EQUIVALENCIAMEMDISC;
DESCRITORJOBUS:DESCRITORJOB;
PLEITURA:LEITURA;
PSPOOL1:SPOOLDEENTRADA;
PCARREGADOR:CARREGADOR;
PPAGINACAO:PAGINACAO;
PESUSUARIO:ESUSUARIO;
PSPOOL2:SPOOLDESAIDA;
PIMPRESSAO:IMPRESSAO;
PCOMPLEMENTAINTERRUPCAO:COMPLEMENTAINTERRUPCAO;
PESCALADOR:ESCALADOR;

BEGIN

INIT USODODISCO,USODAMEMORIA,BCPFISICO,CONTROLBCPS,BUFFERESFI
SICO,CONTROLBUFFERS,CONTROLPAGDISC,CONTROLPAGMEM, EQUIVA
LENCIAMD,DESCRITORJOBUS,PLEITURA(CONTROLBUFFERS,BUFFERES
FISICO);PSPOOL1(CONTROLBCPS,BCPFISICO,CONTROLBUFFERS, BUF
FERESFISICO,CONTROLPAGDISC,USODODISCO);
PCARREGADOR(CONTROLBCPS,BCPSFISICO,CONTROLPAGMEM, EQUIVA
LENCIAMD,USODAMEMORIA,USODODISCO);
PPAGINACAO(CONTROLBCPS,BCPSFISICO,CONTROLPAGMEM,CONTROL
PAGDISC,USODAMEMORIA,USODODISCO,EQUIVALENCIAMD
);

PESUSUARIO (CONTROLBCPS, BCPSFISICO, CONTROLPAGMEM, CONTROL
PAGDISC, USODAMEMORIA, USODODISCO) ;

PSPOOL2 (CONTROLBCPS, BCPSFISICO, CONTROLBUFFERS, BUFFERESFI
SICO, CONTROLPAGDISC, USODODISCO) ;

PIMPRESSAO (CONTROLBCPS, BCPSFISICO, CONTROLBUFFERS, BUFFERES
FISICO) ;

PCOMPLEMENTAINTERRUPCAO (DESCRITORJOBUS, CONTROLBCPS, BCPSFI
SICO, USODAMEMORIA, CONTROLPAGMEM) ;

PESCALADOR (CONTROLBCPS, BCPSFISICO, DESCRITORJOBUS, EQUIVA
LENCIAMD, CONTROLPAGMEM)

END.