Sistema Operacional Simplificado

Ricardo Luís de Freitas

1. Introdução

Um sistema operacional é um programa que age como uma interface entre um usuário e o hardware de um computador. Sua função é gerar um ambiente no qual usuários possam executar programas, utilizando o computador de maneira eficiente. Para isto, o sistema operacional deve atuar como um gerenciador dos recursos de hardware disponíveis e como um programa de controle encarregado de administrar a execução de programas, impedindo erros ou uso impróprio do computador.

A existência de diferentes tipos de sistemas computacionais, como microcomputadores, redes de computadores, sistemas com múltiplos processadores e até máquinas não convencionais, faz com que os sistemas operacionais também difiram variando em estrutura, potência e complexidade, de acordo com os recursos disponíveis que serão gerenciados.

Os principais recursos que um sistema operacional gerência são : processadores, unidades de armazenamento, dispositivos de entrada e saída e dados. Sem tornar inconsistentes os tempos de execução dos programas, um sistema operacional procura maximizar a utilização da UCP, maximizar o número de programas executados num período de tempo, executar cada programa o mais rapidamente possível e minimizar o tempo de resposta entre a máquina e seus usuários interativos.

As operações definidas em um sistema operacional incluem: implementação de uma interface com os usuários do sistema computacional, compartilhamento dos recursos disponíveis entre os usuários, possibilidade de troca de informações, facilidades para a realização de operações de entrada e saída e capacidade de regeneração a partir de situações de erro.

Uma das características mais importantes dos sistemas operacionais é o suporte para multiprogramação. Multiprogramação, uma forma de otimizar o uso do computador, através da execução simultânea de programas e de atividades de E/S, fazendo com que a UCP trabalhe a maior parte do tempo executando programas do usuário.

Um sistema operacional multiprogramado com tempo partilhado mantém residentes em memória diversos programas prontos para execução. Cada programa é executado durante um determinado tempo (time slice), ou até que tenha que esperar pelo atendimento de alguma operação do sistema operacional (tipicamente operações de entrada e saída), passando a UCP para a execução de outra tarefa. Após a realização dessa tarefa, o sistema operacional pode continuar a execução do programa interrompido, ou escalar um novo programa para execução.

Políticas de escalonamento de programas e de tarefas do sistema operacional podem ser baseadas em análises do desempenho dos programas, no tamanho e tempo previsto para suas execuções, na ordem de entrada no sistema, em prioridades pré-

determinadas, ou em outros métodos de avaliação que garantam tempos razoáveis para as suas execuções.

O armazenamento simultâneo de diversos programas, entretanto, exige formas de gerenciamento de memória. Técnicas de compartilhamento permitem que mais programas do que poderiam ser fisicamente armazenados em memória possam estar em execução ao mesmo tempo.

Uma unidade de armazenamento externo (disco, tambor, fita, etc) é utilizada pelo sistema operacional como uma memória secundária onde são armazenados todos os programas completos. O sistema operacional faz um particionamento da memória principal, dividindo-a em pedaços de tamanho fixo (páginas) ou variados, que são compartilhados no armazenamento dos programas em execução.

Sistemas operacionais que fazem uma divisão dinâmica da memória principal transferem um programa inteiro da memória secundária para a principal sempre que este programa for executado. O programa é alocado em um espaço onde ele caiba sobrando o menor ou maior espaço possível, dependendo da política implementada.

Nos sistemas que utilizam paginação, os pedaços dos programas são trazidos para a memória principal à medida em que são necessários para a continuação das suas execuções.

A transferência de um programa (inteiro ou em partes), da memória principal para a secundária, é feita quando ele não está sendo executado e o espaço que estava sendo utilizado para o seu armazenamento , necessário para a execução de outro programa. Os critérios utilizados para escolha das páginas de programas que serão substituídas podem ser baseados em filas, onde a ordem de alocação indica a ordem de substituição; em critérios de utilização, eliminando da memória as páginas que foram utilizadas pela última vez há mais tempo (Least Recently Used - LRU), ou outros métodos.

Ponteiros são mantidos pelo sistema operacional para a identificação do endereço relativo de cada programa na memória. No caso de sistemas que utilizam paginação, uma Tabela de Páginas é armazenada para identificar a presença das páginas dos programas na memória e o endereço de onde elas se encontram.

Com a finalidade de não comprometer o tempo da UCP nas operações de entrada e saída e explorar a possibilidade de paralelismo no funcionamento destes dispositivos, sistemas operacionais utilizam-se de sistemas de 'SPOOL' e buffers na memória. A existência de canais de controle independentes nos dispositivos de E/S permite a utilização de buffers na realização de operações de entrada e saída de dados.

Um canal , um sistema computacional dedicado, utilizado para o controle das operações de E/S. O canal de entrada , ativado para a leitura sempre que houver dados no dispositivo de entrada (terminais, leitora de cartões, fitas, etc). A sua operação é iniciada pela UCP, mas não necessita de supervisão para ser executada.

Enquanto existirem buffers disponíveis para o preenchimento com os dados de entrada, o canal de entrada pode operar, avisando a UCP, através de uma interrupção, sempre que um buffer for completamente preenchido e necessita de tratamento.

Para a realização de operações de saída de dados, o sistema operacional insere os dados em buffers e ativa a operação do canal de saída. O canal de saída é capaz de imprimir os dados sem o controle da UCP, gerando uma interrupção, assim que a impressão for completada, liberando o buffer para reutilização.

Por serem controladas por canais independentes, as operações de E/S podem ser realizadas paralelamente às atividades de controle da UCP na execução de outras tarefas.

O sistema de SPOOL ('Simultaneous Peripherals Operation On Line') introduz a utilização do disco na execução das operações de E/S, armazenando nele os dados que serão lidos pelos programas e os dados que estes irão imprimir. Através deste sistema, a UCP, responsável pelo controle da transferência dos dados do dispositivo de entrada para a memória e dos dados de saída da memória para o dispositivo de saída, não tem que esperar pelas condições de operação desses dispositivos, quando precisa ler ou imprimir dados.

Dados de entrada são lidos do dispositivo de entrada e armazenados no disco em uma área cujo endereço , mantido pelo sistema operacional, para utilização pelo programa a que se destina, quando este executar uma operação de leitura de dados. Da mesma forma, os dados de impressão são gravados em disco e impressos quando o programa terminar sua execução. Utilizando buffers e o sistema de 'Spooling', o sistema operacional evita que a UCP tenha que aguardar pela operação dos canais de E/S, que operam em velocidades relativamente baixas, comparadas à da UCP.

A existência de um sistema de interrupções permite ao sistema operacional compartilhar a utilização da UCP entre os diversos programas residentes e tarefas de controle, alocando-a para a execução dessas atividades e retomando o controle quando interrupções forem geradas. As interrupções indicam também a situação da operação dos canais de E/S.

Para poder alternar a UCP na execução de diversos programas, o sistema operacional armazena as informações relevantes sobre um programa interrompido, podendo restaurá-las quando for retomar a sua execução. Um programa , representado no sistema operacional através de um Bloco de Controle de Programa (BCP), mantido desde a sua entrada no sistema até o final de sua execução.

O BCP , uma estrutura de dados que armazena informações sobre um programa, incluindo o estado atual da sua execução, uma identificação particular a esse programa, a sua prioridade de execução, ponteiros para os recursos e áreas de memória e disco que ele está alocando, o endereço da próxima instrução desse programa que será realizada, informações sobre seu tempo de execução e uma área onde são salvos os conteúdos dos registradores.

Sistemas de alocação de recursos, tratamento de situações de 'deadlock', controle de concorrência e sistemas de proteção a recursos e dados nas unidades de armazenamento, são outras características de sistemas operacionais multiprogramados.

2. Descrição do Sistema Operacional Simplificado

O Sistema Operacional Simplificado , um sistema operacional multiprogramado com execução de programas de usuário em tempo partilhado que opera através de compartilhamento da UCP entre os seus diversos processos concorrentes e os programas de usuários provenientes de terminais.

O sistema operacional está estruturado de acordo com a figura 1, contendo sete processos concorrentes que atuam sobre os periféricos, buffers, disco, memória e outros recursos, como blocos de controle de programas, variáveis de controle, etc. Um conjunto de processos simples e um conjunto de processos especiais compõem os módulos de programas do sistema operacional simulado. Possuindo maior prioridade de execução entre os eventos que concorrem pela utilização da UCP, os processos simples são responsáveis pelo controle da utilização dos recursos do sistema.

O conjunto de processos simples , formado pelos seguintes processos concorrentes :

1 - Leitura
2 - Spool de entrada
3 - Spool de saída
4 - Impressão
5 - Loader
6 - E/S Usuário
7 - Paginação

Os processos de 1 a 4 constituem o Spooling do sistema operacional.

- Leitura : preenche buffers de entrada com o conteúdo das páginas de informação (programas e dados) provenientes dos usuários de um terminal. É executado sempre que existem dados para serem lidos, a leitora não está sendo ocupada e existem buffers disponíveis.

- Spool de entrada : É responsável pela interpretação das páginas de informações lidas dos buffers de entrada e pela colocação dos programas e dados no disco, bem como pela abertura e espaço para impressão, também no disco. Será executado quando houver buffers preenchidos com dados para leitura, espaço livre em disco, BCP's disponíveis e o disco não estiver sendo utilizado.

- Loader (carregamento de programas) : tem a função de alocar um número mínimo de páginas de memória e realizar a carga das páginas necessárias do disco, de

forma a permitir o início da execução do programa. Quando existirem programas residentes em disco que ainda não foram carregados na memória, existir espaço em memória e o disco não estiver sendo acessado, este processo poderá ser executado.

- Paginação : processo encarregado de alocar uma página de memória disponível, para permitir a continuação de um programa, sempre que o sistema detectar falta de página durante a sua execução. Caso a página em falta seja de código do programa, o seu conteúdo é buscado no disco e carregado na página de memória alocada. Se não houver mais páginas disponíveis, o processo de paginação deverá escolher uma página para ser retirada, criando espaço. O processo de paginação proporciona ao sistema operacional um sistema de memória virtual no armazenamento dos programas em execução. Está em condição de ser executado sempre que o sistema operacional detectar a ausência de uma página de programa em memória e o disco não estiver sendo utilizado.

- Entrada e Saída de dados de programas de usuários : permite trazer o conteúdo de páginas de dados do disco para a memória, ou remeter para o disco as páginas de impressão previamente preparadas na memória. Para ser realizado , preciso que o disco não esteja sendo utilizado por nenhum outro processo e haja solicitação de leitura ou escrita por algum programa de usuário.

- Spool de saída : é encarregado da saída de um programa do sistema. Carrega o conteúdo das páginas do disco (programa, dados, resultados) em buffers para impressão. Terminada a execução de um programa, com sucesso ou não, existindo buffers disponíveis e o disco não estando em utilização, este processo ser executado.

- Impressão : sua tarefa é comandar a impressão do conteúdo dos buffers de impressão, sempre que houver buffer preenchido para impressão e o dispositivo de saída não estiver sendo utilizado.

Os processos denominados de escalonador, espera de interrupção, e tratamento de interrupção são os processos especiais do sistema operacional, que proporcionam o seu funcionamento.

Escalonador : é responsável pela verificação de quais eventos estão em condições de serem executados e pela alocação dos recursos necessários para que eles ocorram. Quando não existirem mais processos simples em condição de serem executados, o escalonador tentará escolher um programa de usuário para execução.

Espera de Interrupção : ativado quando não há processos simples ou programas de usuários em condições de serem executados. O processo de espera representa a situação onde o sistema operacional aguarda somente o término da operação de algum canal de E/S que, ao terminar, irá gerar uma interrupção.

Tratamento de Interrupção : fornece ao S.O. condições de retomar o controle dos recursos do sistema. Possui, por isso, a maior prioridade de execução atribuída pelo sistema operacional. É responsável pela análise das necessidades dos programas de

usuários pelo posicionamento dos BCP's nas filas correspondentes e pelo controle do 'status' dos canais de E/S e disco, que habilitam ou bloqueiam a execução dos processos simples.

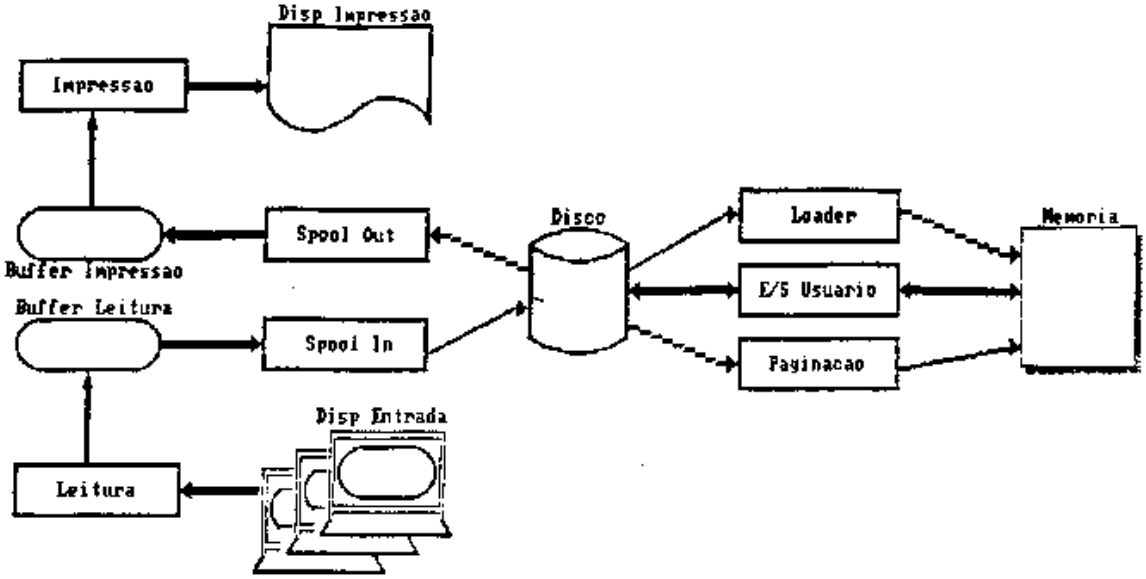


Figura 1 : Estruturação do sistema operacional e seus processos concorrentes

O Sistema Operacional Simplificado corresponde ao seguinte algoritmo :

```
repeat
  EXECUTA_PROCESSOS_SIMPLES;
  if not VAZIA ( FILA_DE_PROGRAMAS_EXECUTÁVEIS )
  then begin
          ESCALA_PROGRAMA_DE_USUÁRIO;
          HARDWARE
        end;
   if not Interrupção
   then ESPERA_INTERRUPÇÃO;  { BUSY WAIT }
  TRATA_INTERRUPÇÃO
until FALSE;  { REPEAT FOREVER }
```

3. Descritor de Programa : BCP

Os programas são representados no sistema operacional através de Blocos de Controle de Programa (BCP), com o seguinte formato :

Identificador do JOB
Endereço da Tab Páginas
ACC
CP
TIMER
TS
FP (nro da página em falta)
Tipo p g em falta (Cod ou Dado)
End do programa no disco
Tamanho do programa
End área de dados no disco
Apontador corrente dos dados
End área impressão no disco
Apontador corrente da impressão
Código de Condição de interrupção
{ área de trabalho }
Ligação para outro BCP

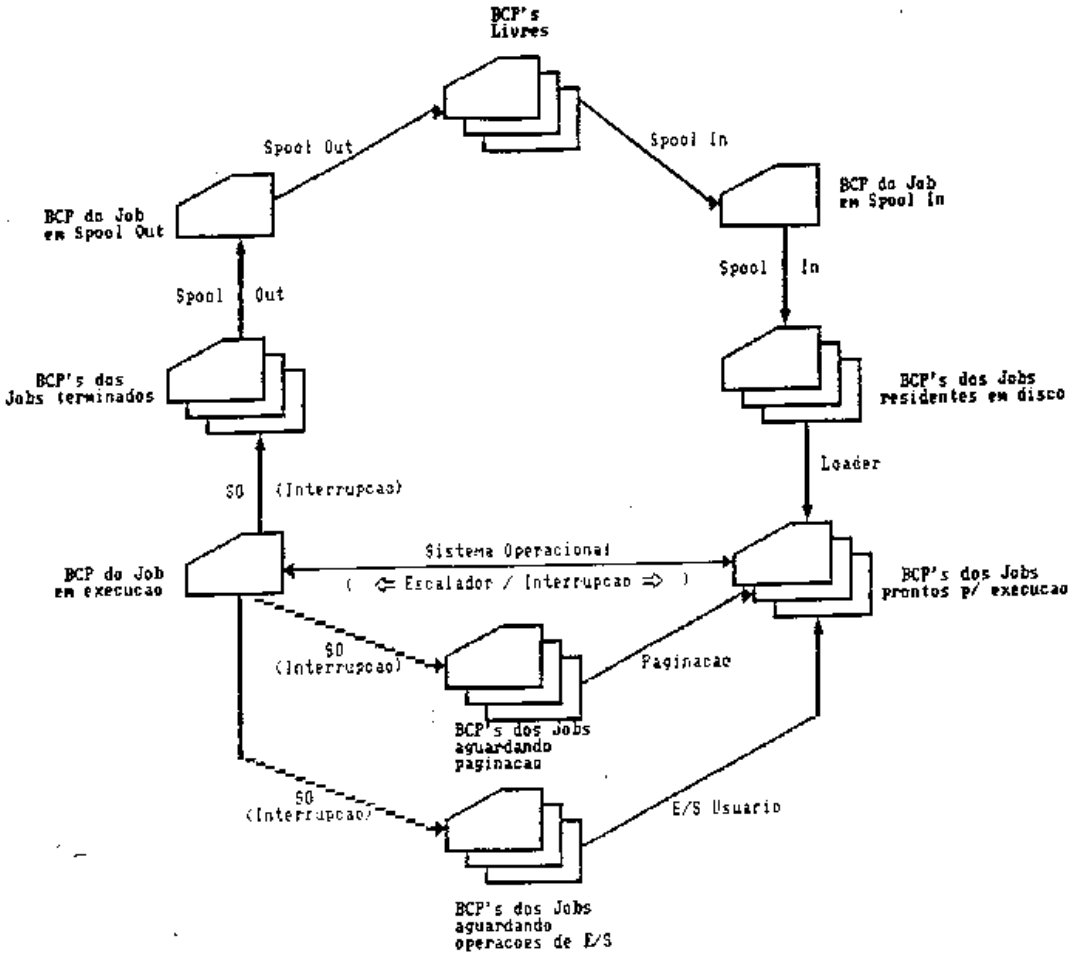


Figura 2 : Fluxo dos BCP's

4. Filas de Programas

Na área de dados do S.O. são criadas filas de Blocos de Controle de Programas (BCP's), que contém os descritores dos programas nos diversos estados de transição durante a execução no sistema operacional.

As filas existentes são as seguintes :

Fila 0 : Fila dos BCP's disponíveis
Fila 1 : Programa em processo de entrada (Spool in)
Fila 2 : Programas residentes em disco
Fila 3 : Programas prontos para execução
Fila 4 : Programa sendo executado

Fila 5 : Programas suspensos aguardando operações de E/S
Fila 6 : Programas suspensos por falta de páginas
Fila 7 : Programas acabados
Fila 8 : Programa em estado de saída

Outras filas podem ainda ser utilizadas para a implementação de sistemas de espera com prioridade, para programas que já entraram no sistema e estão aguardando a operação do Loader, ou filas de programas suspensos por situações diversas.

5. Transição de estados dos programas

Para ser executado, um programa será lido, armazenado em disco e carregado na memória do computador. Uma vez residente em memória, passará então a ser interpretado e executado, até o término normal de sua execução ou até que alguma interrupção fatal ocorra. Terminada sua execução, com sucesso ou não, o programa passará à fase de saída e impressão do seu código e dos resultados das suas operações. O processo de Leitura , responsável pela leitura dos dados do dispositivo de entrada e pelo seu armazenamento em buffers.

Quando o processo de Spool In encontrar um buffer com informações que indicam o início de um novo programa, ele irá alocar um BCP disponível e colocá-lo na fila 1. Conseguindo interpretar corretamente os dados do programa contidos nos buffers de leitura e preencher os dados do seu BCP, o Spool In passará então esse BCP da fila 1 para a fila 2, dos programas residentes em disco. O processo Loader será responsável pela criação da tabela de páginas do programa na fila 2, pelo carregamento da sua primeira página de código na memória, e pela transferência do seus BCP para a fila 3, dos programas prontos para execução.

Quando, no S O, o escalonador escolher um programa para ocupar a UCP, será necessário carregar os registradores com as informações referentes ao estado de execução do programa escalado. Um programa em execução poderá ser interrompido pelos seguintes motivos :

- O programa requer a execução de uma operação de E/S e gera uma interrupção. Ao tratar a interrupção, o sistema operacional colocará o BCP na fila 5, dos programas guardando E/S.

- O pedaço do programa que contém a próxima instrução que dever ser executada não está presente na memória, gerando uma interrupção por falta de página. O tratamento de interrupções irá transferir o BCP para a fila 6, dos programas aguardando paginação.

- Uma interrupção na execução do programa , gerada, devido ao tempo de execução, término normal de execução, ou ocorrência de situações que causariam um erro fatal (proteção de memória, código de operação inválido, ou overflow em operações

matemáticas). O tratamento de interrupção do sistema operacional será responsável pela transferência do BCP do programa para a fila 7, dos programas terminados que aguardam saída do sistema.

OBS : Nas filas 1, 4 e 8 poderá haver somente um programa de cada vez, visto que só poderá existir um programa em fase de entrada, um programa sendo executado e um programa em estado de saída sendo atendidos pelo sistema operacional. Nas demais filas é possível a existência de mais de um BCP's aguardando tratamento.

6. Organização dos programas no disco

Funcionando como uma memória secundária de armazenamento externo, o disco é utilizado para conter os programas em execução no sistema operacional.

Existem três áreas que um programa precisa alocar em disco :

- área de Programa : para armazenamento do código do programa.
- área de Dados: páginas de dados, lidos do dispositivo de entrada, utilizados pelas operações de leitura do programa.
- área de Impressão : espaço vazio que precisa ser reservado para as operações de impressão de dados do programa.

Para a manipulação dessas áreas é necessário que o sistema operacional mantenha ponteiros que indicam o endereço da página corrente de dados, que será lida por um programa e o endereço de onde será impressa a próxima página indicada por uma instrução de impressão dos programas de usuários.

Filas de Buffers de E/S

O S O possui 8 buffers do tamanho das páginas de memória e disco, que são utilizados pelo Spooling do sistema operacional na realização das operações de E/S. Três filas de buffers são utilizadas pelo sistema : fila de buffers aguardando Spool In, fila de buffers aguardando Impressão e fila de buffers livres.

No início da operação do sistema operacional todos os buffers estão disponíveis, na Fila de Buffers Livres. O processo de Leitura aloca buffers livres, para preencher com os dados lidos dos dispositivos de entrada e os insere na fila de buffers aguardando Spool In. Buffers na fila para Spool In são interpretados pelo processo Spool In e liberados para reutilização. O processo de Spool Out , responsável pela saída de um programa do sistema operacional, colocando os dados que serão impressos em buffers disponíveis, inseridos na fila de buffers para impressão.

A impressão dos buffers , comandada pelo processo de Impressão, que envia os dados para o dispositivo de saída e devolve os buffers para a fila de buffers livres, para reutilização.

Problemas de Deadlock na alocação de buffers

Uma das possíveis situações de deadlock previstas e que deve ser evitada pelo sistema operacional é a alocação de buffers pelos processos de Spooling. Em determinado instante, o disco poder ficar completamente cheio, impedindo a ação do processo de Spool In de carregar novos programas dos buffers no disco.

Se, neste instante, todos os buffers estiverem sido preenchidos com dados de entrada, ocorrerá uma situação de deadlock, pois :

- Spool In não pode atuar e por isso mantém buffers preenchidos bloqueados, pois precisa de espaço em disco.

- Spool Out não opera, mesmo quando programas terminar suas execuções, porque precisa preencher buffers com os dados de saída e estes estão bloqueados pelo Spool In. Portanto, não libera espaço em disco.

Uma solução para este problema é nunca permitir que todos os buffers sejam preenchidos para Spool In simultaneamente. Desta forma, garante-se que o Spool Out poder atuar, liberando espaço em disco para o Spool In.

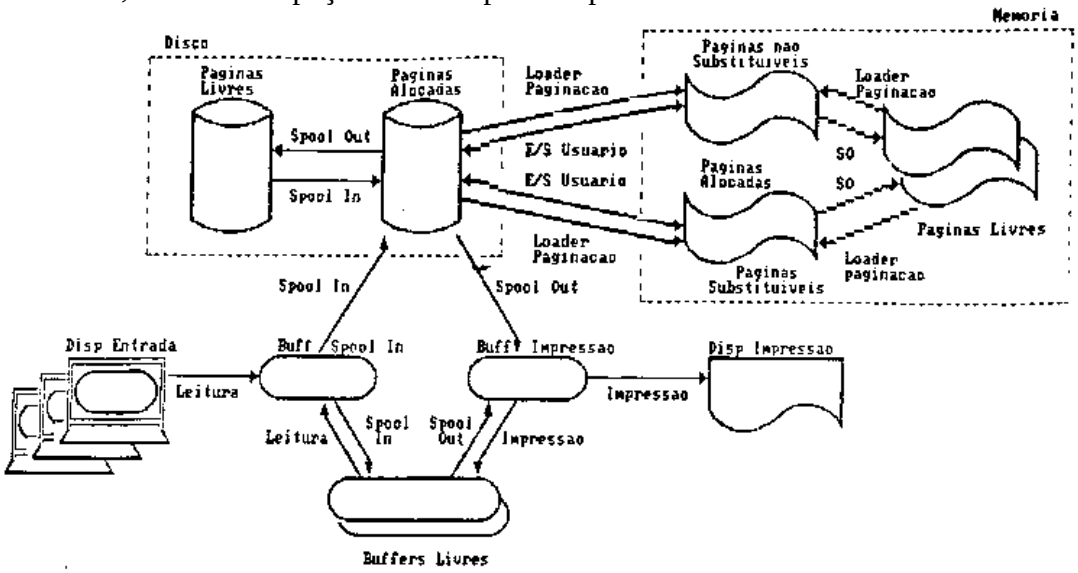


Figura 3 : diagrama de acesso aos recursos do sistema operacional

7. Paralelismo de Execução

Processamento paralelo , uma forma eficiente de processamento de informação que enfatiza a exploração de eventos concorrentes no processo computacional, onde a ocorrência desses eventos concorrentes pode dar-se durante o mesmo período de tempo, durante o mesmo instante de tempo ou durante espaços de tempos sobrepostos.

Em um ambiente com um único processador, o paralelismo de execução explora as seguintes características :

- sobreposição de operações da UCP com operações de entrada e saída que podem ser controladas por canais de processamento autônomo nos dispositivos periféricos.

- uso de multiprogramação e tempo partilhado, com intercalação da UCP na execução de processos e programas.

- balanceamento dos índices de desempenho dos subsistemas do ambiente com o objetivo de eliminar gargalos.

8. Processos Simples

As operações de E/S do sistema, a paginação dos programas na memória e as instruções de E/S dos programas são ações que utilizam os canais de E/S no Sistema Operacional Simplificado e podem ocorrer paralelamente à execução de programas pela UCP.

Embora possam ser executadas de maneira concorrente, essas operações necessitam da supervisão da UCP para ocorrerem. A UCP , encarregada de verificar quando essas ações devem ser executadas, de iniciar suas execuções e de tratar os resultados das suas operações.

Com a finalidade de transformar o controle dessas operações em processos que, assim como os programas de usuários, concorrem pela utilização da UCP, o Sistema Operacional Simplificado implementou os Processos Simples.

Desta forma, os Processos Simples são procedimentos que controlam a utilização dos recursos envolvidos em operações de E/S e armazenamento de dados e programas. Um Processo Simples pode estar ativo, concorrendo pela utilização da UCP, ou inativo. Sua ativação , feita sempre que as condições necessárias para que ele ocorra

são satisfeitas, indicando a necessidade da atuação da UCP no controle de alguma operação de transferência de dados ou programa.

A seguir, são relacionadas as condições necessárias para a realização dos processos.

"Leitura" : - existem dados no dispositivo de entrada
- existe buffer disponível
- o canal de leitura não está sendo utilizado

"Spool In" : - existe buffer na fila de buffers p/ Spool In
- existe espaço disponível em disco
- existe BCP disponível ou em estado de Spool In
- o canal de controle do disco não está em uso

"Loader" - existe espaço na memória (pelo menos 2 páginas : uma para a Tabela de Páginas e uma para a primeira página de código do programa)
- existe BCP na fila dos programas residentes em disco, aguardando carregamento na memória.
- o canal de controle do disco não está em uso

"Paginação" - existe BCP na fila de programas aguardando paginação
- o canal de controle do disco não está em uso

"E/Susuário" - existe BCP na fila de programas aguardando a realização de operações de E/S
- o canal de controle do disco não está em uso

"Spool Out" - existe BCP na fila de programas que já terminaram suas execuções
- existe buffer disponível
- o canal de controle do disco não está em uso

"Impressão" - existe buffer na fila de buffers para impressão
- o canal de controle do dispositivo de saída não está sendo utilizado

A UCP , capaz de identificar quais condições são satisfeitas para a ativação dos Processos Simples. Quando for executado, um Processo Simples irá ativar a operação de um canal de controle de periférico, requerendo leitura ou escrita em um dispositivo. Enquanto espera pelo término da operação iniciada, a UCP não tem mais tarefas para realizar nesse processo e pode então ser alocada para o controle de outras tarefas.

Ao completar a leitura ou escrita que estava executando, o canal gera uma interrupção, indicando à UCP que os dados já foram lidos e podem ser tratados, ou já

foram escritos e que o dispositivo está pronto para reutilização. O Processo Simples que havia sido suspenso volta a ser executado e faz o tratamento da situação.

Para efeito de facilitar a verificação de quais condições são satisfeitas para a realização de cada Processo Simples , útil a implementação de uma tabela que indique a relação entre os recursos e os processos.

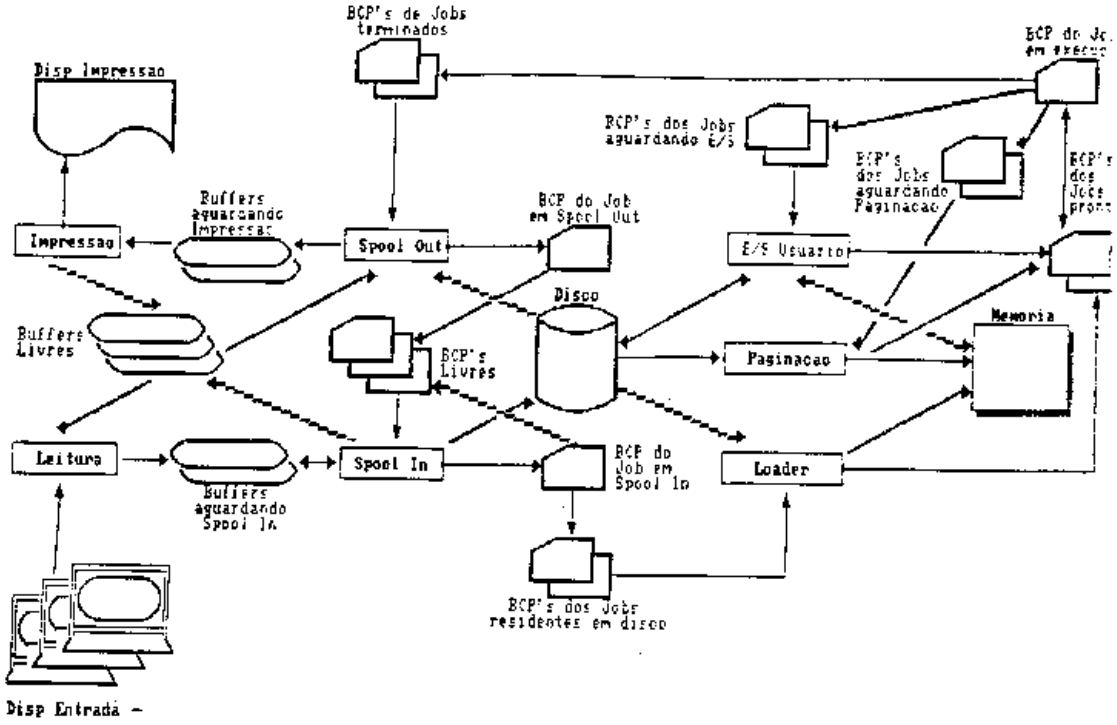


Figura 4 : Organização geral dos processos e recursos do SO e do hardware

9. Tratamento de Interrupções

Num sistema computacional, uma interrupção é um evento que altera a seqüência de execução de instruções por um processador. Uma interrupção é gerada pelo hardware do computador e, quando ocorre, causa a execução dos seguintes passos :

- o controle do processador , retornado para o sistema operacional;
- o sistema operacional salva o estado do processo que estava sendo executado;
- a interrupção é analisada e o sistema operacional passa a executar a rotina de tratamento correspondente.

No Sistema Operacional Simplificado o tratamento de interrupções funciona de acordo com o esquema da figura 5.

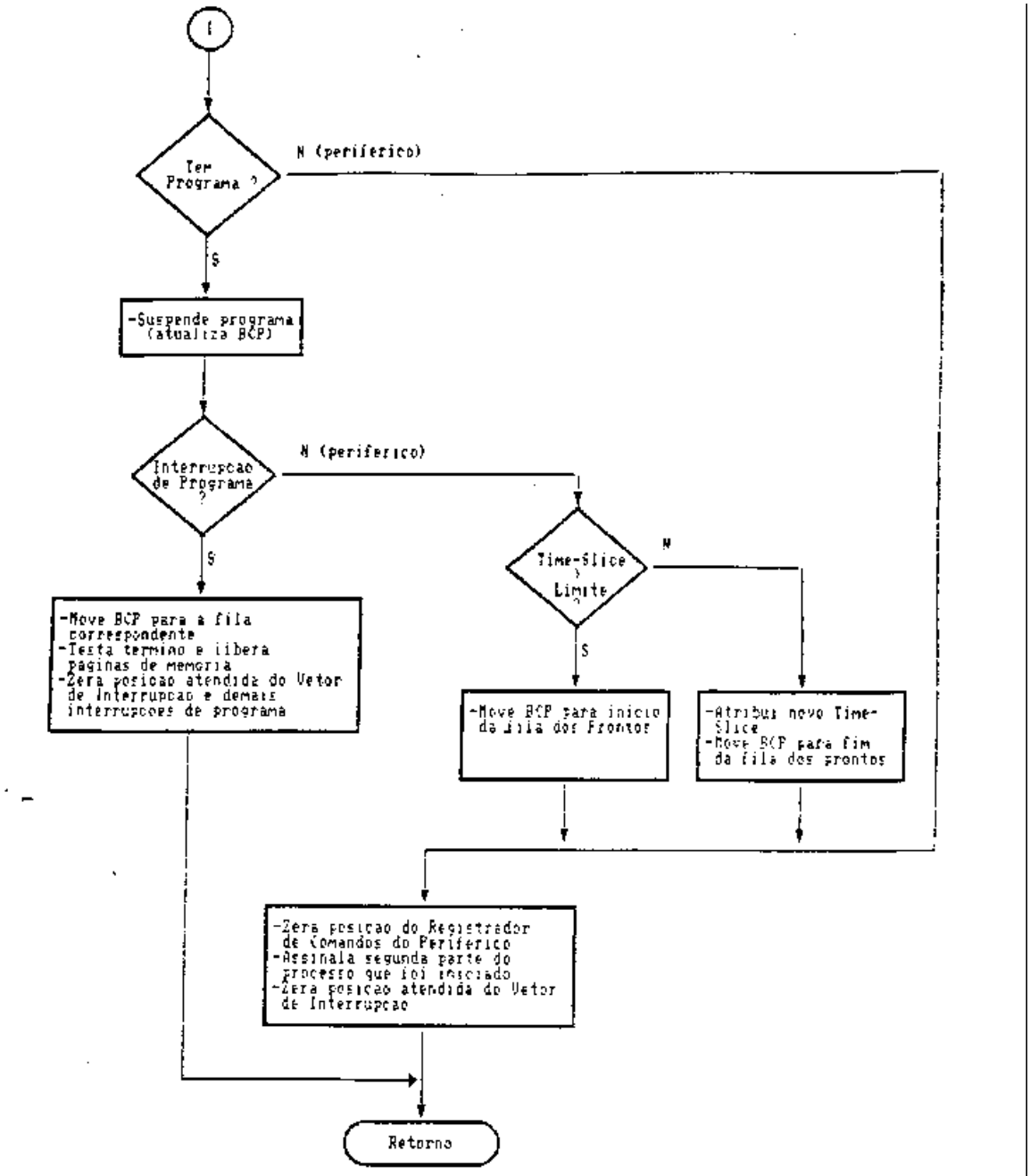


Figura 5 : tratamento de interrupções

Ao fazer o tratamento de uma interrupção de periférico, o sistema operacional é capaz de identificar qual Processo Simples havia comandado a sua operação e deve ser reativado. O tratador de interrupção libera o periférico para reutilização, mas é necessário que o Processo Simples que iniciou a sua operação seja reativado para tratar

outros recursos envolvidos, como buffers ou BCP's que devem ser liberados ou transferidos para outras filas existentes, ou dados que foram lidos e devem ser tratados.

Na implementação do Sistema Operacional Simplificado pode ser utilizado um vetor que indique quais Processos Simples, suspensos enquanto aguardavam pela operação de periféricos, podem ser reativados depois do tratamento de interrupções de periféricos.

Vetor de Condição de Processos Simples - Segunda parte

Leit	Sp In	Load	Pag	E/SUs	Sp Out	Impr
(0/1)	(0/1)	(0/1)	(0/1)	(0/1)	(0/1)	(0/1)

Cada elemento do vetor , assinalado pelo tratamento de interrupção e zerado pelo escalonador do sistema, após a escolha do processo simples que será reativado para o término de sua execução.

10. Hardware

O hardware do Computador Simplificado (CS), implementado por software através de procedimentos e estruturas de dados, simula um computador, contendo os recursos necessários para sustentar multiprogramação (Figura 6).

Seus componentes incluem :

- uma unidade central de processamento (UCP).
- memória principal de acesso rápido.
- uma unidade de armazenamento externo (disco).
- registradores específicos e de uso geral
- contador do tempo para execução de programas (TIMER).
- contador da fatia de tempo da UCP para a execução de cada programa (TIME SLICE).
- relógio (CLOCK).
- unidades de entrada e saída de dados controladas por canais autônomos.
- vetor de interrupções

Construído com a finalidade de proporcionar um ambiente que restringisse os detalhes da implementação do sistema operacional somente às suas características funcionais, o hardware simulado abstrai detalhes de baixo nível na sua construção. Desta forma, a unidade básica do seu sistema de armazenamento ('palavra') é especificada através de uma estrutura de dados.

Uma palavra do CS , um registro contendo três campos para valores inteiros :

Palavra :

C1	C2	C3

Ao armazenar uma palavra de instrução, os campos de uma palavra terão os seguintes significados :

C1 : Código da Operação
C2 : Página referida pela instrução
C3 : Deslocamento dentro da página referida (palavra)

Quando representar um dado, somente o primeiro campo da palavra será relevante, armazenando o valor do dado :

C1 : Valor do dado armazenado
C2 : - (Irrelevante)
C3 : - (Irrelevante)

Para facilidade de desenvolvimento do sistema simulado, foi definida uma memória do usuário, composta por 256 palavras e sujeita a controles e manipulação por parte do sistema operacional. Para a implementação de um sistema de paginação e memória virtual, entretanto, o sistema operacional irá estruturá-la em blocos de 8 palavras, formando 32 páginas.

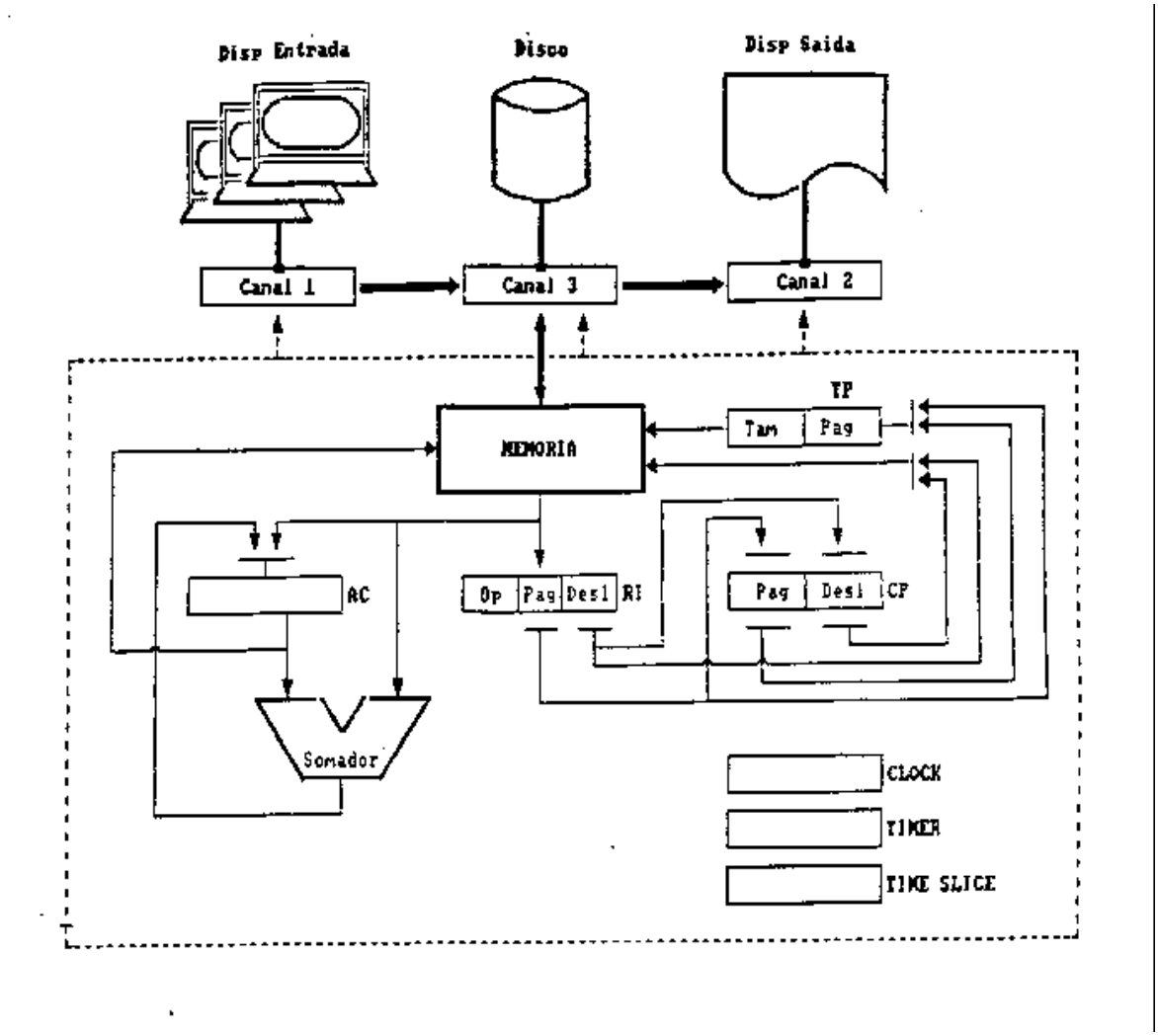
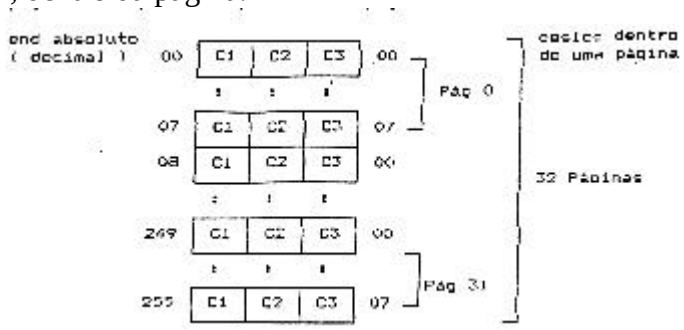


Figura 6 : organização do hardware

Desta forma, um endereço qualquer v lido XYZ se refere à página XY e palavra Z, dentro da página.

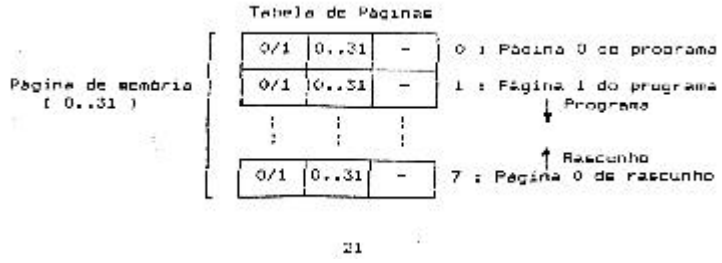


Com a finalidade de permitir que mais programas possam ser posicionados na memória simultaneamente para execução, o sistema operacional implementa um sistema de endereçamento relativo, com memória compartilhada através de paginação. Esse sistema de paginação faz com que os programas sejam tratados por partes, de modo que seus códigos não necessitam estar completamente carregados em memória durante as suas execuções e nem estejam alocados de maneira seqüencial, ocupando espaços contíguos na memória.

Os programas são divididos em páginas, do mesmo tamanho que as páginas de memória, que são carregadas em espaços disponíveis da memória ou sobrepondo outros pedaços de programa à medida em que são necessárias para a continuidade da execução dos programas. Para poder identificar a presença das páginas dos programas na memória e a sua eventual localização, o sistema operacional monta então uma Tabela de Páginas para cada programa, instalada em uma página comum de memória.

Cada palavra da Tabela de Páginas representa a presença e o endereço efetivo de uma página do programa do usuário na memória, de acordo com os seguintes campos :

C1 : Indica presença ou ausência de uma pág do prog na memória
C2 : Endereço efetivo da página na memória (se estiver presente)
C3 : (não utilizado)



21

Desta forma, para obter o endereço efetivo de onde se encontra uma palavra de instrução ou de dados de um programa, o sistema operacional segue o seguinte procedimento :

- Identifica a página de memória que contém a Tabela de Páginas do programa.
- Pesquisa nessa página, na palavra correspondente à página buscada, a presença e o endereço efetivo da página.

O sistema operacional mantém uma Tabela de Páginas para cada programa em execução. Por ser instalada em uma página comum de memória, entretanto, o tamanho dos programas é limitado a, no máximo, 8 páginas, incluindo código e páginas de rascunho.

A Tabela de Páginas (TP) representa a presença e localização tanto das páginas de código quanto das páginas de rascunho reservadas. A representação das páginas de código apresenta uma correspondência direta entre o número da linha da TP

e o número da página. Desta forma, a presença e localização da página 0 do programa do usuário , indicada pela linha 0 da sua TP, a página 1 pela linha 1 e assim sucessivamente.

Para a representação das páginas de rascunho, entretanto, a correspondência ocorre de maneira invertida : a página 0 de rascunho , representada pela linha 7 da TP, a página 1 pela linha 6, e assim por diante, no sentido contrário às linhas que representam as páginas de código.

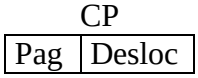
Para o cálculo da posição da TP que indica a localização de uma página de rascunho, a expressão (7 - Nro Pag) pode ser utilizada. Assim : Página 0 => 7-0 = 7
Página 2 => 7-2 = 5, etc.

A Unidade Central de Processamento do CS foi projetada com o intuito de prover suporte de hardware necessário à execução de vários programas com tempo partilhado e paginação na memória do usuário.

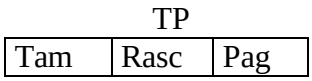
Os principais registradores da UCP são :

ACC (Acumulador) : Um registrador, para valores inteiros, para finalidades gerais.

_CP_ (Contador de Programa) : Um registrador de dois campos para valores inteiros, que fazem referência à página e à palavra da próxima instrução de programa a ser executada.



_TP_ (Tabela de Páginas) : Um registrador que contém o endereço da Tabela de Páginas de um programa na memória (0..31), o tamanho do programa em páginas (0..7), e o número de páginas de rascunho reservadas.



_FP_ (Falta Página) : Um registrador para um campo de valor inteiro que armazena o número da página em falta no programa que estava sendo executado.

CK (Clock) : relógio da UCP (valor inteiro).

CK

TIMER : Registrador para um campo de valor inteiro que é inicializado com o tempo previsto para a execução de um programa e é decrementado à medida em que ele é executado.

Timer

TS (TIME SLICE) : Contador da Fatia de Tempo atribuída pelo sistema operacional a um programa, para utilização da UCP na sua execução.

TS

A unidade central de processamento opera através de um conjunto simplificado de instruções de m quina que permite a elaboração de programas de usuário, envolvendo manipulação de posições de memória, operações aritméticas, controle de desvio e parada de programas e operações de E/S. As instruções dividem-se em dois grupos principais : as que fazem referência à área de código (instruções de desvio de execução e fim de programa) e as que manipulam a área reservada para rascunho (demais instruções).

As seguintes operações são implementadas :

HLT : Indica o fim de processamento de um programa.

Instrução		
0	---	---

_RD XY 0_ : Lê a página corrente da área de dados do programa no disco e carrega o seu conteúdo na página XY da área de rascunho do programa.

Instrução		
1	XY	0

PRN XY 0 : Imprime, na área de impressão do disco alocada pelo programa, o conteúdo da página XY da área de rascunho desse programa.

Instrução		
2	XY	0

LD XYZ : Carrega no acumulador (ACC) o conteúdo da palavra Z da página XY da área de rascunho do programa sendo executado. No caso do sistema simulado, a operação carrega no acumulador somente o conteúdo do primeiro campo da palavra de dados.

Instrução		
3	XY	Z

STR XYZ : Armazena o conteúdo do acumulador na palavra Z da página XY da área de rascunho do programa. Atribui o valor do primeiro campo da palavra.

Instrução

4	XY	Z

SUB XYZ : Atribui ao acumulador o valor correspondente ao valor que este continha, subtraído do valor armazenado no primeiro campo da palavra Z da página XY da área de rascunho do programa do usuário.

Instrução

5	XY	Z

ADD XYZ : Idem à subtração, atribuindo ao acumulador o resultado da soma ao invés da subtração.

Instrução

6	XY	Z

JMP XYZ : Atribui um novo valor para o Contador de Programa, do programa em execução, que passa a ser executado a partir dessa posição.

CP.PAG = XY
CP.DESL = Z

Instrução

7	XY	Z

JNG XYZ : O conteúdo do acumulador é verificado e, caso contenha um valor negativo, o valor de XYZ é atribuído ao CP. Caso o acumulador não seja negativo, o programa continua sua execução a partir da próxima instrução em relação ao endereço em que se encontra.

Se ACC < 0
então (CP.PÁG = XY
CP.DESL = Z)

Instrução

8	XY	Z

MUL XYZ : Atribui ao acumulador o valor correspondente ao valor que este continha, multiplicado pelo valor armazenado no primeiro campo da palavra Z da página XY da área de rascunho do programa do usuário.

Instrução

9	XY	Z

DIV XYZ : Atribui ao acumulador o valor correspondente ao valor que este continha, dividido pelo valor armazenado no primeiro campo da palavra Z da página XY da área de rascunho do programa do usuário.

Instrução

9	XY	Z

Assembly Interativo

RDI XYZ

Rea**D I**nterativo – Faz uma leitura de teclado para uma página de dados específica. Caso a página de dados não esteja alocada, um erro de proteção de memória é gerado.

RDI	Página	Deslocamento

PRI XYZ

PRint **I**nterativo – Executa uma leitura na página de dados especificada e envia para o canal de saída do monitor.

PRI	Página	Deslocamento

NOVAS INSTRUÇÔES

Mnemônico: **LDB**

	Op1	*Op2*
LDB		

Descrição:
Carrega registrador B com o valor no deslocamento *Op2* da página *Op1*.

Mnemônico: **STB**

	Op1	*Op2*
STB		

Descrição:
Armazena o conteúdo do registrador B no deslocamento *Op2* da página *Op1*.

Mnemônico: **MAB**

MAB	-----	-----

Descrição:
Move o conteúdo do acumulador Acc para o registrador B.

Mnemônico: **MBA**

MBA	-----	-----

Descrição:
Move o conteúdo do registrador B para o acumulador Acc.

Mnemônico: **XCH**

XCH	-----	-----

Descrição:
Permuta o conteúdo do registrador B e o acumulador Acc.

Mnemônico: **LIA**

LIA	*Op1*	-----

Descrição:
Carrega acumulador Acc imediato com o valor *Op1*.

Mnemônico: **LIB**

LIB	*Op1*	-----

Descrição:
Carrega registrador B imediato com o valor *Op1*.

Mnemônico: **SAB**

SAB	-----	-----

Descrição:
Subtrai acumulador Acc do registrador B, guardando o resultado em B. Ou seja, faz B = B-Acc.

Mnemônico: **SBA**

SBA	-----	-----

Descrição:
Subtrai registrador B do acumulador Acc, guardando o resultado no acumulador. Ou seja, faz Acc= Acc-B.

Mnemônico: **ABA**

ABA	-----	-----

Descrição:
Adiciona o conteúdo do registrador B no acumulador Acc.

Mnemônico: **AAB**

AAB	-----	-----

Descrição:
Adiciona o conteúdo do acumulador Acc no registrador B.

Mnemônico: **INA**

INA	-----	-----

Descrição:
Incrementa o conteúdo do acumulador Acc.

Mnemônico: **INB**

INB	-----	-----

Descrição:
Incrementa o conteúdo do registrador B.

Mnemônico: **AIA**

AIA	*Op1*	-----

Descrição:
Adiciona imediato *Op1* no acumulador Acc.

Mnemônico: **AIB**

AIB	*Op1*	-----

Descrição:
Adiciona imediato *Op1* no registrador B.

Mnemônico: **ANA**

ANA	*Op1*	*Op2*

Descrição:
Operação bitwise (bit-a-bit) *AND* entre o acumulador Acc e o valor contido no deslocamento *Op2* da página *Op1*. Resultado fica armazenado no acumulador.

Mnemônico: **ORA**

ORA	*Op1*	*Op2*

Descrição:
Operação bitwise (bit-a-bit) *OR* entre o acumulador Acc e o valor contido no deslocamento *Op2* da página *Op1*. Resultado fica armazenado no acumulador.

Mnemônico: **XRA**

XRA	*Op1*	*Op2*

Descrição:
Operação bitwise (bit-a-bit) *XOR* entre o acumulador Acc e o valor contido no deslocamento *Op2* da página *Op1*. Resultado fica armazenado no acumulador.

Mnemônico: **ANB**

ANB	-----	-----

Descrição:
Operação bitwise (bit-a-bit) *AND* entre o acumulador Acc e o registrador B. Resultado fica em Acc.

Mnemônico: **ORB**

ORB	-----	-----

Descrição:
Operação bitwise (bit-a-bit) *OR* entre o acumulador Acc e o registrador B. Resultado fica em Acc.

Mnemônico: **XRB**

XRB	-----	-----

Descrição:
Operação bitwise (bit-a-bit) *XOR* entre o acumulador Acc e o registrador B. Resultado fica em Acc.

Mnemônico: **NEA**

NEA	-----	-----

Descrição:
Nega (inverte bit-a-bit) o conteúdo do acumulador Acc.

Mnemônico: **NEB**

NEB	-----	-----

Descrição:
Nega (inverte bit-a-bit) o conteúdo do registrador B.

Mnemônico: **IVA**

IVA	-----	-----

Descrição:
Efetua complemento de 2 (inverte) no conteúdo do acumulador Acc.

Mnemônico: **IVB**

IVB	-----	-----

Descrição:
Efetua complemento de 2 (inverte) no conteúdo do registrador B.

Mnemônico: **JAB**

JAB	*Op1*	*Op2*

Descrição:
Se Acc >= B, o programa continua sua execução a partir do deslocamento *Op2* da página *Op1*.

Mnemônico: **JBA**

	Op1	*Op2*
JBA		

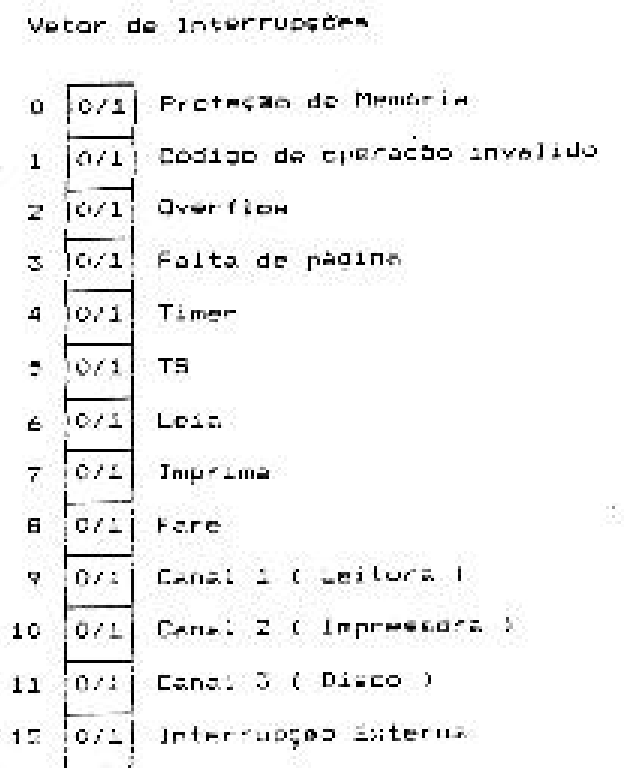
Descrição:
Se B >= Acc, o programa continua sua execução a partir do deslocamento *Op2* da página *Op1*.

O CS contém ainda um registrador de interrupções que indicam anomalias ou sinais de alerta gerados pelo hardware em determinadas condições. O registrador de interrupções , implementado através de um vetor de 12 posições com campos que podem assumir os valores 0 ou 1. O valor 1, em um elemento desse vetor, representa a ocorrência da interrupção a que corresponde. O valor 0 indica a não ocorrência de uma interrupção.

Os elementos de 0 a 3 do vetor correspondem à interrupções de programa, que são geradas em decorrência das suas execuções.

O valor 1 no elemento 0 do vetor indica que o programa tentou acessar uma posição de memória fora da sua área de código ou de rascunho ou tentou ler ou imprimir dados no disco numa área fora do espaço que havia alocado.

O elemento 1 do vetor recebe o valor 1 sempre que um programa tentar executar uma operação cujo código não é reconhecido pelo sistema operacional.
Se, durante a operação de uma instrução aritmética, houver overflow no valor armazenado no acumulador, o elemento 2 do vetor de interrupções receberá o valor 1.



Caso, no cálculo do endereço efetivo de uma posição de memória referenciada por um programa, sua tabela de páginas indique que a página não está presente na memória, uma interrupção por falta de página é gerada, atribuindo-se o valor 1 ao elemento 3 do vetor. TIMER e TS são interrupções de tempo e são geradas quando esses registradores assumem o valor 0. Leia, Imprima e Pare são interrupções do sistema e indicam a necessidade de realizações de ações do sistema operacional. As interrupções dos canais de controle dos periféricos indicam ao sistema operacional o estado da Leitora, da Impressora e do Disco e são ativadas sempre que estes dispositivos completam a realização de uma operação de Entrada/Saída. Interrupções externas, geradas pelo escalonador ou por um usuário do sistema, podem ser geradas e são identificadas pelo elemento 12 do vetor de interrupções.

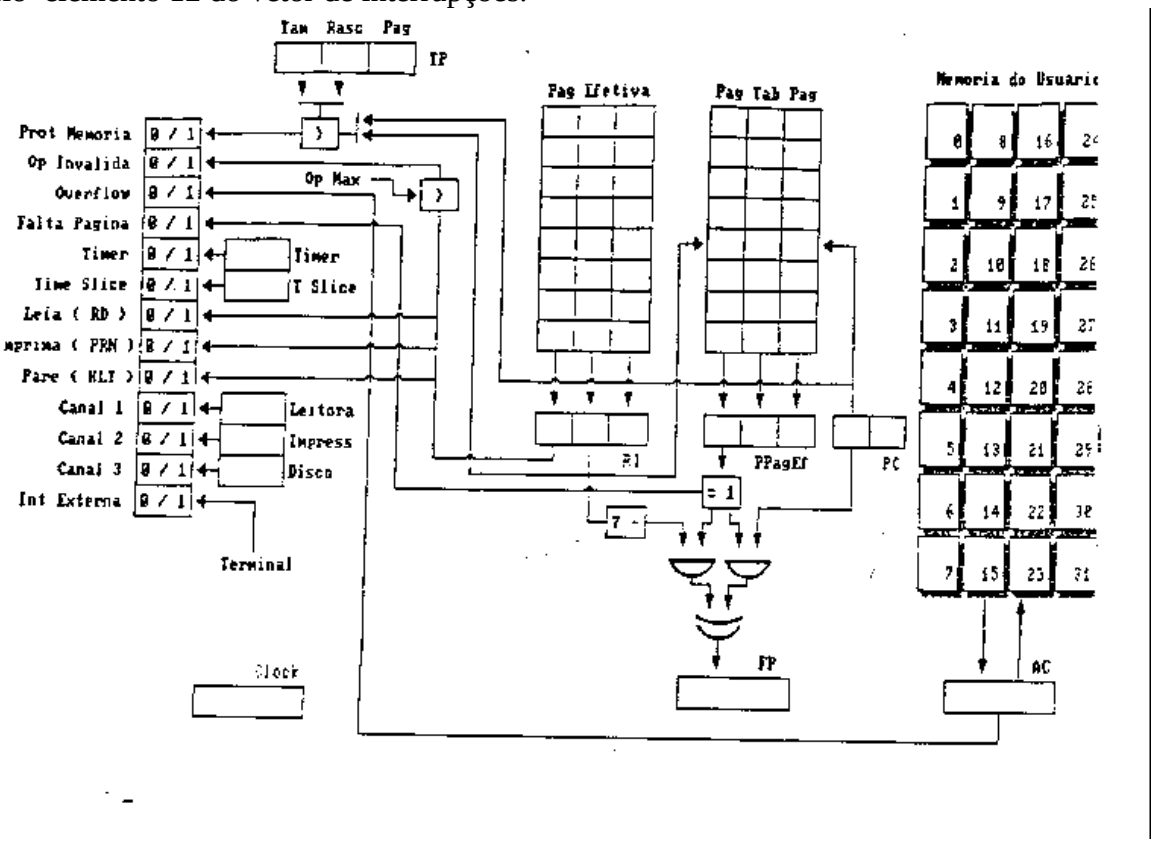


Figura 7 - Hardware com vetor de interrupções

Em qualquer condição de interrupção, se houver um programa em execução, ele é interrompido logo após a realização da instrução que estava sendo tratada e o controle é desviado para a rotina de tratamento de interrupções do sistema operacional. As informações referentes à situação dos registradores do hardware e do estado da execução do programa são salvos no seu BCP e podem ser restauradas quando ele voltar a ser executado.

As interrupções por proteção de memória, código de operação inválido, overflow e timer são consideradas fatais e sua ocorrência durante a execução de um programa causa o término forçado de sua execução e a sua saída do sistema.

11. Hardware simulado

A simulação do hardware leva em conta o funcionamento da UCP, realizando busca e execução de instruções e o funcionamento dos periféricos, executando suas operações principais. Durante a simulação, são verificadas as condições de interrupção e é atualizado o tempo definido para a execução dos diversos módulos do sistema. A simulação do hardware ocorre de acordo com o algoritmo colocado a seguir.

```
repeat
 BUSCA_INSTRUÇÃO;  { FETCH }
  if not INTERRUPÇÃO
  then begin
          EXECUTA_INSTRUÇÃO;
          SIMULA_PERIFÉRICOS
        end;
until INTERRUPÇÃO;
```

Simulação dos periféricos

O Computador Simplificado suporta um dispositivo de entrada de programas dos usuários, uma impressora e um disco ou tambor, cada um ligado ao seu respectivo canal de controle.

Um registrador de comando, contendo três campos, representa o estado de operação dos canais de E/S.

RC

Leit	Impr	Disc

0: Disponível
1: Em operação

A simulação dos periféricos ocorre de maneira bastante simplificada, uma vez que são considerados somente a execução das operações e o tempo estabelecido para isto. Ao iniciar sua operação (ativação do periférico), a variável associada ao periférico , atualizada com o valor do tempo padrão gasto na operação do dispositivo e, após o decorrer desse tempo, que pode ser utilizado para execução de atividades da UCP e operação de outros canais, uma interrupção é gerada, indicando que a operação foi desempenhada.

Um registrador de três campos para valores inteiros, associados aos canais de E/S, , utilizado para simular o tempo de operação dos periféricos.

T

Leit	Impr	Disc

Especificações dos periféricos:

Periférico	Canal	Registro	Tamanho	Vel (Tempo Operação)
Leitora	1	1 página	-	10 unidades
Impressora	2	1 página	-	10 unidades
Disco	3	1 página	256pag	3 unidades

12. Organização dos programas de usuário - CS / SOS

Para efeito de se verificar a organização de um programa de usuário é importante que se observe algumas características básicas do sistema de armazenamento do sistema operacional :

- uma palavra , formada por três campos para valores inteiros e pode armazenar uma instrução ou um dado
- uma página , composta de 8 palavras
- o espaço de memória destinado aos programas de usuários contém 32 páginas
- para cada programa , mantida uma tabela de páginas, que ocupa uma página comum de memória
- as instruções em geral referem-se à palavras, exceto as instruções LEIA e IMPRIMA que fazem referência à páginas.
- qualquer endereço referenciado por uma instrução ou pelo contador de programa (CP) , sempre relativo a uma página dentro da área de código ou da área de rascunho do programa do usuário, e não a uma posição efetiva na memória. Para encontrar o endereço efetivo à que se refere a instrução ou o CP, o sistema operacional deve consultar a tabela de páginas desse programa.
- para a representação das páginas de código do programa a correspondência entre as linhas da Tabela de Páginas e a numeração das páginas é direta. Na representação das páginas de rascunho a correspondência se dá na forma invertida, ou seja, 7 – o número da página indicada pela instrução.
- Cada palavra da tabela contém informação sobre a presença ou não da página correspondente na memória e, caso esteja presente, o endereço de onde ela se encontra.

Desta maneira, poderia-se pensar que o tamanho próximo do programa de usuário capaz de ser submetido ao sistema seria de 64 instruções (8 páginas). Ocorre por,m que, para a realização de entrada e saída através das instruções LEIA PÁG e IMPRIMA PÁG, o

programa deve prever páginas para leitura e para impressão de dados reduzindo, portanto, o tamanho próximo do programa.

Assim, se for suficiente para o usuário trabalhar com uma página de dados e uma página de impressão, executando instruções que manipulem as palavras dessas páginas, o programa ficará limitado a 6 páginas (48 instruções). Poderão aparecer casos onde uma única página satisfaz as necessidades de E/S, sendo que a montagem da página de impressão é feita sobre a página de dados, que não tem mais utilização, ou casos onde são necessárias mais de uma página para leitura ou para impressão, quando os dados manipulados são em número maior que a capacidade de armazenamento de uma página, ou quando a geração de dados para impressão ocorre em posições seqüenciais, exigindo uma área maior que uma página. Deve-se considerar que o programa deverá prover uma área de trabalho onde ele possa armazenar os valores intermediários gerados e utilizados pelo programa.

Desta forma :

Tam máx do programa = 64
- nro de palavras da área de trabalho
- (8 * nro pg simultâneas de leitura)
- (8 * nro pg simultâneas de impressão)

Essas limitações no tamanho do programa do usuário se devem à simplicidade do tratamento da Tabela de Páginas. O espaço para a área de trabalho pode ser reservado através da alocação de páginas de rascunho utilizando-se comandos da Linguagem de Controle de Programas do sistema operacional.

Linguagem de Controle de Programa

O sistema operacional trabalha sob o comando de linhas de controle preparadas pelos usuários, que podem configurar o sistema e pedir a execução de programas, passando automaticamente de programa para programa com um mínimo de tempo e intervenção de operadores.

Através da Linguagem de Controle de Programas, o sistema operacional controla e direciona as operações de E/S dos programas, controla o armazenamento e a alocação de espaços em disco, determina as prioridades de execução dos programas e dispõe sobre a configuração do sistema.

A Linguagem de Controle de Programas do Sistema Operacional Simplificado possui a seguinte sintaxe para as linhas de controle :

* Job { Início de um novo programa }
  Identificação do programa
  Tempo previsto para execução
  Tamanho do programa
  Número de páginas de rascunho
  Número de páginas de impressão

* Prog { Início das linhas de código }

Linhas de programa { Instruções }

[* Dado { Início das linhas de dados }

Linhas de dados] { Dados }

* Fim

A interpretação das linhas de controle , executada pelo processo de Spool de Entrada do sistema operacional e opera segundo o autômato de transição de estados ilustrado na figura 8.

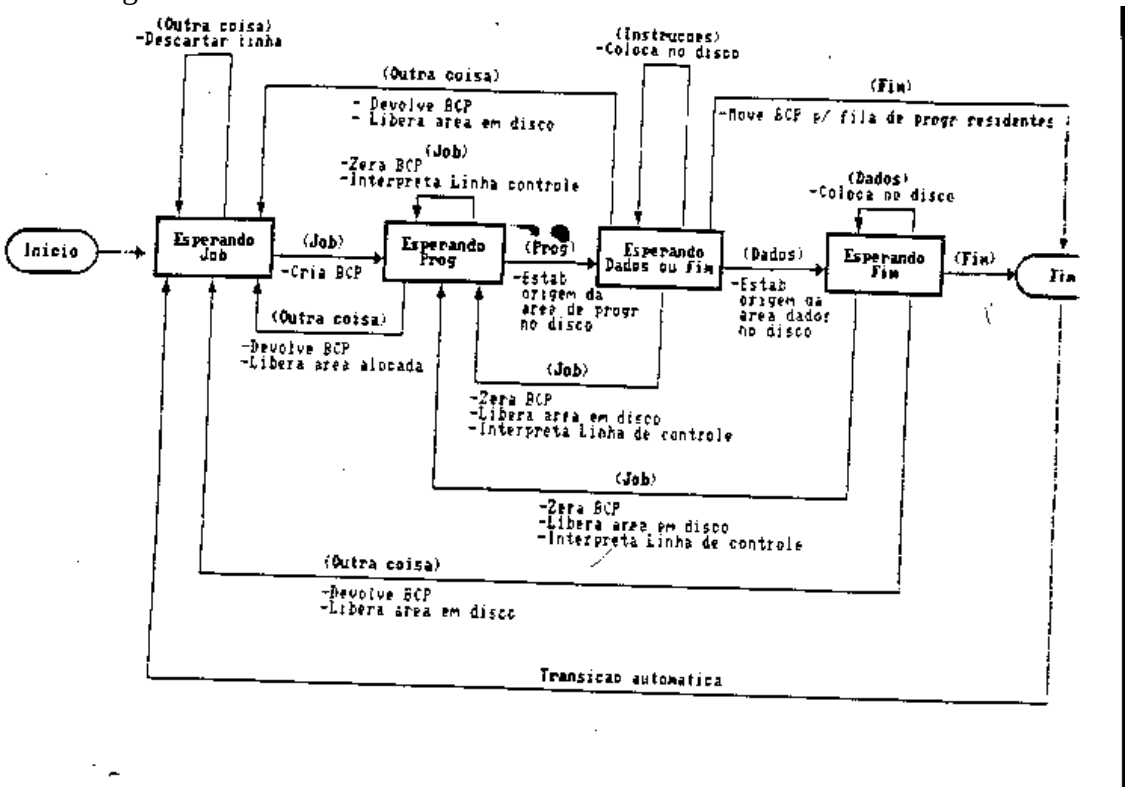


Figura 8 : Spool de entrada - autômato de interpretação das linhas de controle