Sistema operacional para processamento paralelo

Cavalli, E.

Zabeu, M. C.

Este artigo expõe as características de um sistema operacional desenvolvido no CPqD - Telebrás. É fundamentalmente um sistema operacional para aplicações de tempo real, implementado de forma a suportar diretamente estruturas hardware com multiprocessadores e sistemas distribuídos. Dá-se particular enfoque nas características que possibilitam o uso do mesmo para processamento paralelo.

1 - Introdução

Neste artigo é descrito o sistema operacional PP-SO/P desenvolvido no Centro de Pesquisa e Desenvolvimento da Telebrás. É um sistema de escopo geral que será utilizado nas aplicações baseadas no equipamento "Processador Preferencial" também desenvolvido no CPqD.

O PP-SO/P é fundamentalmente um sistema operacional para aplicações de tempo real, implementado de forma a suportar diretamente estruturas hardware com multiprocessadores e sistemas distribuídos.

Antes de descrever as características do PP-SO/P será feita uuma breve descrição da estrutura hardware suportada pelo sistema operacional. A seguir, serão examinadas algumas soluções do PP-SO/P a problemas específicos do processamento paralelo e também serão analisados alguns problemas que ainda devem ser solucionados.

2 - Estrutura hardware

O processador preferencial (PP) é um microcomputador de 16 bits constituído das seguintes placas:

- UCP: unidade central de processamento, baseada no iAPX 286, com 512 Kbytes de memória RAM, EPROM de até 64 Kbytes, interfaces seriais, entre outras características;

- DIS: controlador de discos flexíveis e rígidos (ST 506);

- SER: controlador de interfaces seriais (8);

- FMG: controlador de fita magnética e interface GPIB;

- MEM: memória RAM com capacidade de 128 a 2048 Kbytes;

- UCC: unidade de comunicação padrão ETHERNET.

Diferentes configurações de placas podem ser interligadas por um barramento de 16 bits de alta velocidade (até 10 Mbytes/seg). No barramento podem ser colocadas uma ou mais UCP's e placas controladoras, num total de até 16 placas.

Um exemplo de estrutura suportada pelo PP-SO/P é representado na figura 1:

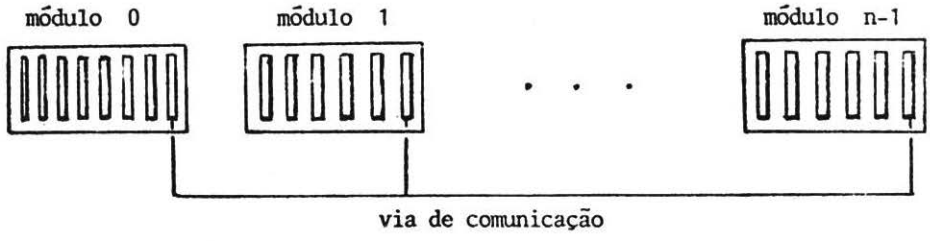


figura 1 - estrutura suportada pelo PP-SO/P

Os módulos são interligados por uma via de comunicação que deve ser adequada às necessidades do sistema. O PP-SO/P é implementado de forma a permitir diferentes vias de comunicação. Atualmente existem dois tipo de placas UCC, uma que utiliza 2 controladores Intel 82586 (conexão tipo ETHERNET) para obter duas vias de comunicação (uma é utilizada como reserva fria) e uma outra mais simples e mais rápida que gerencia contemporaneamente várias vias físicas de forma a aumentar velocidade e confiabilidade.

Cada módulo é constituído de um sub-bastidor cuja configuração mínima é de uma UCP e uma UCC. Um exemplo de configuração do módulo é representado na figura 2:

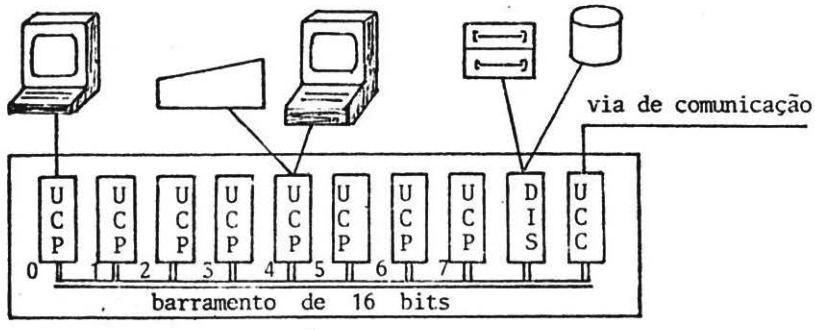


figura 2 - configuração de um módulo

As placas UCP utilizam o barramento unicamente quando acessam elementos exteriores às placas. Não existindo possibilidade de comunicação direta entre UCP's os acessos externos são limitados às operações de E/S que utilizam placas externas e aos acessos à memória (MEM). A MEM é utilizada para permitir a comunicação entre as UCP's do mesmo módulo.

O código executado pelas UCP's deverá obrigatoriamente residir na memória interna às UCP's para evitar o bloqueio de barramento com consequente degradação do desempenho global do módulo.

2 - Definições básicas para o entendimento do PP-SO/P

2.1 - Conceito de instância

A instância é o identificador do menor elemento software concorrente gerenciado pelo PP-SO/P. Cada elemento identificado por uma instância é executado "concorrentemente" no sistema. A concorrência de execução pode ocorrer em processadores (UCP) diferentes ou no mesmo processador. A concorrência dentro do mesmo processador ocorre conforme regras de prioridade. Depende também das ações executadas pelos elementos que provocam a suspensão temporária dos mesmos e de eventos externos (ações executadas por outros elementos) que provocam a reativação de elementos suspensos.

A instância tem o formato descrito na figura 3. A descrição deste formato fornece algumas importantes características do PP-SO/P.

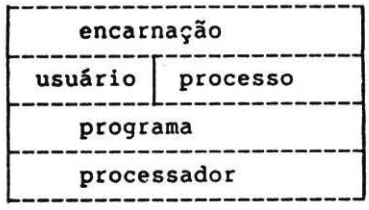


figura 3: formato de instância

O campo "processador" indica em qual processador o elemento se encontra, isto é, o PP-SO/P pode suportar diretamente o processamento distribuído. O campo "usuário" define o dono do elemento. O campo "programa" identifica o produto software. O programa é a unidade mínima que pode ser carregada e por consequência pode também ser considerada como a unidade básica para efeito de produção e manutenção. O programa carregável é o

resultado de um ou mais módulos gerados por compilação. O programa é carregado de forma indivisível em um processador e é formado de um ou mais "processos".

Um processo pode ter uma ou mais encarnações, isto é, o código de um processo porde ter mais que uma ocorrência concorrente. A criação de (encarnações de) processos é feita dinamicamente pela primitiva "START" e é análoga à função "fork" do UNIX.

Observe-se que todos os elementos são concorrentes. Há concorrência entre programas e concorrência dentro do programa se o mesmo tiver mais de um processo (a palavra "processo" é utilizada por brevidade como sinônimo de "encarnação de processo").

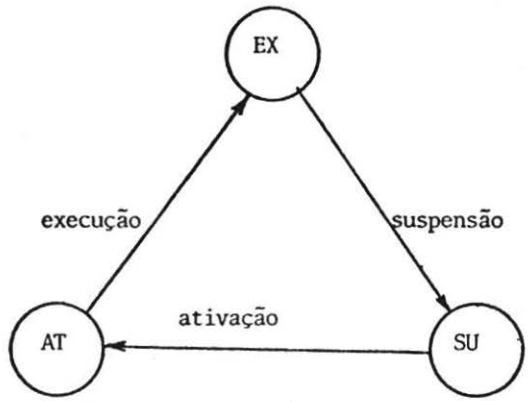
Para facilitar a compreensão da descrição a seguir são aqui definidos os estados de um processo:

- execução: estão sendo executadas suas instruções;

- ativo: na fila de espera para ser executado;

- suspenso: parado à espera de uma ocorrência externa (sinal, evento, liberação de região, temporização, fim de E/S, entre outras).

Os estados são mostrados na figura 4:



3 - Características do PP-SO/P

O PP-SO/P é um sistema operacional para sistemas distribuídos que utiliza o iAPX 286 no modo "protegido" que, além de permitir o acesso a uma maior quantidade de memória (16 Mbytes), permite isolar e proteger completamente programas e garante a integridade do próprio sistema operacional contra acessos indevidos. Esta característica é condição necessária para um sistema operacional multiusuário e multiprogramado como o PP-SO/P. Algumas características do iAPX 286 modo protegido que podiam prejudicar o mecanismo de proteção e isolamento foram proibidas.

O PP-SO/P foi desenvolvido para atender às necessidade de controle de processos em tempo real em aplicações que requerem alto grau de confiabilidade. Além disso, o sistema operacional tem um sistema de E/S sofisticado para gerenciar "arquivos" em sistema distribuído e para fornecer os serviços normalmente utilizados nas operações de operação e manutenção. Estes serviços, que incluem a gerência de periféricos padrões, levam o sistema a ter características gerais. Para suportar estas diferentes necessidades o PP-SO/P foi desenvolvido de forma modular e facilmente configurável.

A seguir serão descritas as características do PP-SO/P divididas em funções básicas, sistema de E/S, suportes a depurações e ao processamento distribuído.

3.1 - Funções básicas

A parte básica do PP-SO/P é relacionada com a linguagem CHILL (CCITT High Level Language) do CCITT (the International Telegraph and Telephone Consultative Comittee). Muitas funções não previstas no CHILL foram colocadas para atender a necessidades reais.

As funções básicas são as seguintes:

- escalonador de processos: são suportadas 3 classes de prioridades (A, B, C), cada uma com 8 níveis internos de prioridade. Os processos da classe C operam em divisão de tempo.

- conexão de processo a um relógio ou a uma interrupção (primitiva START);

- alocação de memória partilhada para E/S (ALLOCATESM);

- ativação de processo (primitiva START) definindo classe e nível de prioridade. O mesmo processo pode ser ativado mais de uma vez criando várias encarnações do mesmo;

- terminação de processo (STOP);

- alocação e desalocação de memória local (ao programa) ou global (entre programas) (ALLOCATE, ALLOCATEG, TERMINATE);

- temporizações. Existem três classes de temporização de valores configuráveis;

- identificação da instância corrente (THIS);

- auto suspensão de processo temporizada ou agendada (SLEEP);

- mecanismo de sincronização entre processos de um programa (EVENT, CONTINUE);

- implementação de acesso controlado a regiões (LOCK, UNLOCK);

- mecanismo de sincronização e comunicação entre processos independentemente da localização dos mesmos (SEND, RECEIVE, RECEIVEALL). O endereço de destino de um sinal tem o formato de instância descrito no item "2".

- envio de sinais temporizados ou agendados. A função fornecida pode ser representada da seguinte forma:

```
SEND sinal TO x AFTER 60 minutos EVERY 2 minutos FOR 5 horas
SEND sinal TO x AFTER 3/abril EVERY 8:31 horas  UNTIL 8/maio
```

- definição de data e hora (SETDATETIME);

- mudança de horário (CHANGETIME);

- define, cancela ou obtém equivalências de símbolos e/ou nomes lógicos (ASSIGN, DEASSIGN, GETASSIGN);

- instalação de usuário (via sinal);

- carregamento de programa (via sinal).

A possibilidade de conexão de processos a ocorrências de relógio ou a interrupções físicas permite a implementação de "rotinas" temporizadas de alta prioridade e a implementação pelo usuário do controle de placas específicas da aplicação sem necessidade de modificar o sistema operacional.

Para implementar o controle de placa o programa deverá ter o privilégio necessário para executar instruções que acessam o espaço de E/S. Na implementação do controlador o usuário não precisa executar o tratamento da interrupção na UCP mas unicamente as operações relativas à placa.

O tratamento de interrupções internas à UCP é executado por primitivas do PP-SO/P que permitem:

- habilitar ou desabilitar interrupções (START, INTCONTROL, RETNOSCHED, RETSCHED);

- terminar o tratamento da interrup@~ão interna à UCP (RETNOSCHED, RETSCHED). A primitiva RETSCHED provoca o reescalonamento dos processos de forma a permitir que um processo de classe mais prioritária (que aquela do processo em execução) eventualmente ativado pelo tratamento da interrupção seja tratado imediatamente.

Note-se que um processo de uma classe x não pode ser interrompido por outro processo da mesma classe mesmo que este tenha um nível de prioridade maior.

O segundo entra em execução somente quando o primeiro processo executar uma operação que provoque a sua suspensão (espera de sinal, chamada de E/S, acesso a região ocupada, espera de evento, etc.).

Esta restrição tem o objetivo de facilitar o controle da coerência dos dados em aplicações de tempo real e de reduzir o número de reescalonamentos. A restrição não é prejudicial ao funcionamento porque nas aplicações em tempo real as transições (código executado até chegar a um novo ponto de suspensão) não são demoradas.

Os processos com transições demoradas devem ser executados na classe C, onde a execução é interrompida a intervalos periódicos de tempo; o processo é colocado no fim da fila dos processos ativos do seu nível de prioridade, sendo reativado após um tempo determinado (divisão de tempo).

3.2 - Sistema de E/S

As principais características do sistema de E/S são:

- implementação modular estruturada;

- execução concorrente em tempo real das operações de E/S que fisicamente podem ser realizadas contemporaneamente;

- configurabilidade em linha de uma unidade;

- sistema de arquivos UNIX;

- acesso a arquivos e periféricos (os periféricos são tratados como arquivos) residentes em outros processadores;

O sistema de E/S é dividido modularmente em três partes:

a) Tratador das primitivas. Implementa a interface entre o software de aplicação e as demais partes do sistema de E/S;

b) Gerenciador do sistema de E/S. Gerencia o sistema de arquivos e executa as operações de E/S mais complexas. É formado por um conjunto de processos que operam concorrentemente com os demais processos. Existe uma encarnação de gerenciador para cada sistema de arquivos ativo no sistema;

c) Controladores Software. Implementam a interface entre o hardware e as demais partes do sistema de E/S.

A interface dos controladores software com o sistema de E/S é padronizada por meio de duas rotinas: IOWAIT e IODONE. A IOWAIT coloca o pedido de execução de uma operação na fila do controlador software. As operações que podem ser executadas pelos controladores software são:

- inicia placa;
- inicia canal físico;
- configuração unidade;
- ligar/desligar unidade;
- resetar unidade;
- atualizar operações pendentes;
- posicionar;
- leitura de dados;
- escrita de dados;
- aborta operações pendentes;
- formata unidades;
- cópia entre unidades;
- escritas especiais;
- leitura de estado da placa.

O canal físico, neste contexto, é uma via de acesso concorrente fornecida pela placa. Por exemplo, na placa DIS tem-se um canal para disco flexível, um para disco rígido, um para entrada serial e outro para a saída serial porque a placa suporta operações concorrentes nessas vias.

Nem todas as operações são significativas para todos os tipos de controladores software, mas devem ser implementadas como rotinas vazias (somente com IODONE).

A rotina IODONE é chamada pelo controlador quando for terminada a operação requisitada.

No PP-SO/P existem controladores software para interfaces seriais da UCP, das placas SER, FMG e DIS, para disco flexível, disco rígido, vídeo e teclado PC-compatíveis, para disco dual e para fita magnética.

O sistema de E/S executa concorrentemente as suas operações. Para garantir a concorrência a cada encarnação de controlador software é associada uma fila de pedidos e, analogamente, cada gerenciador de E/S (um gerenciador é associado a um sistema de arquivos presente no sistema) tem uma fila dos seus requerimentos. Um controlador software tem tantas encarnações quantos canais físicos ele serve. Por exemplo, o controlador software da entrada serial terá tantas encarnações quantas são as entradas seriais correntemente ativas no sistema. Este número é variável dinamicamente.

O sistema permite a configuração de unidades periféricas sem a necessicade de desligar o sistema. Exemplos típicos deste caso são a instalação de discos rígidos com especificações não padronizadas e a conexão de novos terminais via interfaces seriais.

O sistema de arquivos utilizado é aquele do UNIX com as características de proteção e as modalidades de acesso do mesmo.

Os arquivos e os periféricos remotos, isto é, controlados por outros processadores, podem ser acessados como qualquer outro arquivo ou periférico local desde que a função de E/S remota esteja configurada e que a comunicação para isso já tenha sido estabelecida. A aplicação pode fazer isso de forma muito simples uma vez que o "nome do processador" é parte do nome do arquivo.

A seguir tem-se uma lista das principais primitivas de E/S disponíveis para o usuário:

- abre/fecha arquivo (ASSOCIATE, DISSOCIATE). Note-se que o arquivo pode ser uma unidade física;

- cria/cancela arquivo (CREATE, DELETE);

- cria/cancela diretório (CREATEDIR, DELETEDIR);

- muda nome de arquivo (MODIFY, RENAME);

- define modalidade de acesso ao arquivo (CONNECT);

- obtém informações sobre o tipo de arquivo e condições de uso (SEQUENCIABLE, VARING, WRITEEXCLUSIVE, GETUSAGE, OUTOFFILE, etc.)

- leitura de registro (READRECORD);

- escrita de registro (WRITERECORD);

- liga/desliga arquivos (LINK, UNLINK);

- fornece nome de arquivo (GETNAME);

- define/fornece diretório corrente (SETDIR, GETDIR);

- modifica atributos de arquivo (MODIFYATTR);

- fornece estado da última operação sobre um arquivo (GETIOSTATUS);

monta/desmonta sistema de arquivos (MOUNT, UNMOUNT);

- procura arquivo cujo nome corresponde a uma cadeia contendo "*" e "?" (FINDMATCH);

- acesso direto a console (READC, WRITEC, READLINE, WRITELINE, INPUTREADY) - leitura, escrita de caracteres, leitura e escrita de linha e verificação de caractere na entrada.

3.3 - Suportes à depuração

O PP-SO/P dispões de algumas funções para desenvolver ferramentas para teste de programas. Todas estas funções são controladas e operam por meio de sinais, isto é, são acessíveis a partir de qualquer processador.

As funções são:

- rastreamento: esta função permite visualizar sinais e ocorrências de eventos gerando sinais de rastreamento nos seguintes casos:

* liberação de sinal;

* eventos do tipo: vencimento de temporização, liberação de regiões, primitivas "START" e "STOP", etc.

As condições de rastreamento podem ser colocadas a nível de programa, processo ou encarnação. Pode ser definido um gatilho de parada, um gatilho de partida e condições de filtro. O rastreamento de sinais pode ser executado com propagação simples ou condicionada.

- depuração: esta função permite colocar condições de parada no código (breakpoints) e na ocorrência de determinadas situações (região, evento, temporização, SEND, RECEIVE, falhas, etc.). As condições de parada podem ser colocadas a nível de programa, processo e encarnação. Quando a condição for encontrada o programa inteiro é parado e poderá ser continuado por meio de um comando. Existem também comandos para leitura e escrita em

memória.

- informação: esta função fornece informações sobre a estrutura e dados utilizados pelo sistema operacional. Por exemplo, configuração do sistema, número de usuários, dados do programa, etc.

- emulação: esta função envia os sinais sem destino para um determinado processo do usuário. É utilizada para emular software nas fases de teste.

- falha: o usuário pode definir um processo para onde serão enviados os sinais de falha.

O PP-SO/P fornece também um mecanismo de "watch-dos" a nível de transição, isto é, a duração de cada execução sequencial de qualquer processo do sistema pode ter uma temporização associada.

Para evitar bloqueios devidos a mau funcionamento do hardware dos periféricos todas as operações de E/S podem ser temporizadas.

4 - Processamento paralelo

O PP-SO/P tem características que o transformam em uma base ideal para o processamento paralelo em estruturas hardware onde existe baixo acoplamento entre os processadores. Este tipo de processamento paralelo é chamado também de processamento distribuído.

As principais características são:

- mecanismo de sincronização e comunicação entre elementos software (encarnações de processos) independente da localização dos elementos. A sincronização e a comuniçação é feita usando as seguintes primitivas:

  * SEND, que envia um sinal para um elemento com determinada instância;

  * RECEIVE, que recebe determinados sinais.

Um exemplo de utilização deste mecanismo é:

```
RECEIVE ptrinstância, time-out = 5ms

(sinal x):

     bloco 0 de ações

(sinal y):

     bloco 1 de ações
```

```
(sinal z):
     IGNORE                    ;sinal ignorado
(sinal t):
     SAVE                      ;salva sem tratar
(ALLOTHERS):
     bloco 3 de ações          ;ações executadas
                               sobre todos os
                               demais sinais
ON (time-out):
     bloco 4 de ações          ;ações executadas
                               no caso de time-out
```

O "ptrinstância" é o ponteiro de uma locação do tipo "instância" onde será colocada a instância de quem enviou o sinal.

* RECEIVEALL: recebe qualquer sinal.

- instalações remotas de usuário: um determinado usuário em um processador A pode-se instalar também em um processador B. O PP-SO/P não suporta instalações múltiplas do mesmo usuário no mesmo processador.

- carregamento remoto de programas: um usuário de um processador A pode carregar programas em um processador B desde que seja instalado em B e que a instalação tenha sido executada por requerimentos originados a partir do processador A.

- acesso de arquivos remotos: um programa de um processador A pode acessar arquivos (e periféricos) controlados por um processador B de forma transparente, isto é, utilizando as mesmas primitivas que possibilitam os acessos aos arquivos (e periféricos) locais. Para que isso seja possível já deve ter sido estabelecida uma interconexão para E/S entre processadores. A operação de conexão é feita por meio da primitiva CONESR (conexão de E/S remota) e a desconexão por meio da primitiva DESCONESR (desconexão de E/S remota). A desconexão pode ser executada unicamente quando não houver mais arquivos remotos em uso.

As operações de E/S remotas introduzem uma perda de desempenho em relação aos acessos locais, mas as vantagens oferecidas em termos de custos e de sistema são grandes. A perda de desempenho será tanto menor quanto mais rápida for a via de comunicação e quanto maior for a quantidade de dados envolvidos

em cada operação de leitura ou escrita.

- gerência de programas remotos: o sistema operacional de um processador mantém informações sobre os programas por ele carregadoss nos demais processadores. Isso implica em que, por exemplo, no fim do carregamento e na terminação de um programa carregado em B pelo processador A, o sistema operacional de A deve ser informado.

- todos os mecanismos de suporte à depuração operam por via de sinais, isto é, podem ser controlados a partir de qualquer processador. As informações sobre os elementos de um processador A podem ser obtidas via sinais por um programa no processador B. Isso facilita a implementação de um controlador centralizado dos recursos de todo o sistema.

- existem mecanismos que permiter avaliar de forma relativamente simples, ainda que não com muita precisão, a percentagem de carga de processamento e a quantidade de memória disponível.

5 - Conclusões

O sistema operacional PP-SO/P fornece mecanismos de processamento paralelo perfeitamente adequados para aplicações que assumem a tarefa de controlar diretamente a alocação dos programas nos processadores do sistema, especialmente aqueles programas que constituem-se de partes de um único processamento e que são correlatos por meio de troca de dados e de condições de sincronização.

A operação de divisão da aplicação em partes que podem ser colocadas em paralelo deve ser executada em um nível alto da análise do problema. O processamento paralelo com baixo acoplamento não permite implementar com eficiência o paralelismo de microoperações de rápida execução.

A divisão do programa em partes concorrentes deve ser executada mais pelo analista que pelo programador. Isso reduz um pouco a dificuldade em ter pessoas capacitadas para resolver este tipo de problema. Ou seja, reduz o número de pessoas que necessitam deste conhecimento. Por outro lado cabe ressaltar que uma nova abordagem de programas sob o enfoque do processamento paralelo é, aparentemente, menos difícil que a passagem da programação sequencial normal para a programação em tempo real. Aliás, esta abordagem nada mais é que uma utilização mais intensiva da programação em tempo real.

Uma outra solução do problema que deve ser estudada é a utilização de linguagens que facilitem a atividade de partição da aplicação. Sendo a partição executada em um nível alto a linguagem também deverá ser de alto nível no sentido de permitir

a separação lógica entre as partes do programa.

O PP-SO/P não possui atualmente um gerenciador de alocação dos programas nos processadores. Para a implementação deste gerenciador os principais problemas a serem considerados são:

- a criação de um mecanismo de controle de recursos, configuração e carga de todos os processadores do sistema;

- fornecimento pela aplicação das partes (programas) em que o problema foi dividido e as características dos mesmos que sejam úteis para a alocação:

  * utilização de memória comum (devem ser colocados no mesmo processador);

  * quantidade de intercomunicação;

  * quantidade de utilização de periféricos de E/S;

  * indicação de alocação forçada em um determinado processador;

  * indicação do tipo de paralelismo possível entre os programas;

  * características úteis para avaliação do tipo de programa em relação ao paralelismo;

- como atualizar, ou informar, os programas em relação às instâncias dos programas com os quais existem correlações de dados e/ou sincronização.

A análise e a solução destes problemas é condição necessária para ter um sistema operacional orientados por aplicações de tipo geral onde haja características de multiprocessamento e multiusuários. Neste contexto o usuário normal não conhece a estrutura lógica da máquina e deve ser de responsabilidade do sistema operacional a distribuição de tarefas entre os vários processadores.