

PCS 3216 - Sistemas de Programação

Aula 22 – Sistemas Operacionais

Introdução

- Nesta aula, é feita uma rápida visita ao tema dos **Sistemas Operacionais**, tópico da maior importância no âmbito dos Sistemas de Programação.
- Neste contexto, a exemplo do caso de Linguagens e Compiladores, Sistemas Operacionais também são **estudados em disciplina a parte** no nosso curso.
- Por isso, não são aqui estudadas as minúcias metodológicas, de arquitetura ou de implementação dos sistemas operacionais, dando-se preferência ao viés para a **área dos Sistemas de Programação**, para caracterizar o papel dos Sistemas Operacionais como **coordenador das atividades executadas** no software de um sistema computacional.

INTRODUÇÃO AOS SISTEMAS OPERACIONAIS

Algumas Razões para o Estudo dos S.O.

- Facilitar a tarefa de **projetar ou modificar** um sistema operacional para uso específico em uma aplicação.
- Para a maioria das instalações de computadores, a seleção de um Sistema Operacional adequado é de grande importância, portanto o profissional da área deve saber **especificar o Sistema Operacional**.
- O Sistema Operacional é o veículo pelo qual o usuário interage com o sistema. Logo, devem-se **escolher suas características** para facilitar o usuário.
- **Conceitos e técnicas** estudadas para a construção do Sistema Operacional frequentemente são **utilizados em outras aplicações** de software.

Sistema Operacional

- O **sistema operacional** é o conjunto dos módulos de Software que regem a **utilização dos recursos** do sistema, **resolvem conflitos**, **simplificam o uso** do computador e procuram **melhorar o seu desempenho** global.
- **Recursos** são elementos de software e/ou hardware que são necessários para que um programa do usuário possa ser executado.
- São geralmente tratados como **Recursos** pelo sistema
 - **Processadores**;
 - **Memória** (principal ou secundária);
 - Dispositivos de **Entrada e Saída**;
 - **Arquivos**;
 - **Programas e Bibliotecas** do Sistema

Importância do Sistema Operacional

- É imenso o montante anual de gastos em computação, e desses recursos, mais de 70% são dirigidos ao **software**. Isto justifica a preocupação com o bom uso desse recurso, fazendo-a produzir mais rapidamente resultados melhores. Os Sistemas Operacionais são ferramentas de que o programador pode dispor para **melhorar sua eficiência**.
- O **Sistema Operacional é o intermediário** entre o usuário e a máquina. Um Sistema Operacional bem projetado facilita a utilização da máquina pelo programador, ao contrário de um Sistema Operacional mal feito, que pode dificultar ao usuário a utilização dos recursos computacionais.
- O Sistema Operacional deve providenciar meios de **aumentar a eficiência global** do computador, mesmo que para isso programas particulares possam ser eventualmente prejudicados. O Sistema Operacional deve fazer o possível para **manter ocupados e produtivos** o mais perto dos 100% do tempo todos os **recursos disponíveis**.

O PAPEL DO SISTEMA OPERACIONAL NOS SISTEMAS DE PROGRAMAÇÃO

O S.O. no Sistema de Programação

- Os diagramas mostrados nos próximos slides esquematizam o **papel desempenhado pelos diversos módulos** do sistema de programação e a interação usual entre eles.
- O **sistema operacional** está colocado em uma posição que simboliza sua **ascendência sobre todos esses módulos**, e também sobre os programas do usuário, pois ele desempenha o papel de regente, que determina quando, onde e qual dos programas pode ou deve executar alguma operação, coordenando-os para que possam ser executados harmonicamente.

Sistema Operacional no Sist. de Programação



Componentes do Sistema Operacional

Programas de sistema

Interpretador de comandos do S.O.

Máquinas virtuais

Editores de texto

Expansores de macros

Carregadores Absolutos

Relocadores-ligadores

Montadores

Interpretadores

COMPILADORES

Serviços de sistema

Segurança e privacidade

Tratamento de interrupções

Virtualização de recursos

Sistema de arquivos

Administração de dispositivos

Administração de processos

Administração de Memória

Interface gráfica

Comandos do sistema

Chamadas de sistema

SISTEMA OPERACIONAL

PROCESSADOR REAL (hardware)

Componentes de um S.O.

- Compõem o sistema operacional duas classes:
 - **Programas de sistema**, que abrangem todos os módulos já estudados do sistema de programação, incluindo também um **interpretador de comandos** do S.O.
 - **Bibliotecas de sistema**, disponibilizadas para serem requisitadas pelos programas do usuário para executarem atividades que dependam de recursos privilegiados, apenas acessíveis a programas do sistema e não aos do usuário.
 - **Serviços de sistema**, que podem ser acionados por comandos do sistema operacional ou então pela execução de instruções privilegiadas de **chamadas de sistema**, que provocam interrupções, cujo tratamento promove o acionamento desses serviços diretamente por um programa em execução.

Algumas visões do sistema operacional

- O Sistema Operacional pode ser encarado de diversas maneiras, conforme o ponto de vista de quem o observa.
- As visões usualmente encontradas (há outras) são:
 - **Administrador dos Recursos** do Computador
 - **Coordenador da Transição dos Estados** dos Processos
 - **Extensão do Hardware** (Máquina Hierárquica Estendida)
 - **Prestação de Serviços** aos programas e ao usuário
 - **Interpretação de linguagem de comandos** do operador
 - Servidor e executor de **chamadas de sistema**
 - Ambiente de **comunicação e sincronização** dos processos
 - Moderador de **políticas, direitos e prioridades** de processos
 - Servidor de **Entrada/Saída** no sistema
 - Ambiente de **tratamento de exceções e interrupções**
 - Ambiente de **virtualização de recursos**
 - Elemento de **suporte à segurança** no sistema

S.O. COMO ADMINISTRADOR DOS RECURSOS DO SISTEMA

Administração dos Recursos do Computador

- Para fazer com que os **recursos** (memória, processadores, Entrada/Saída, informações etc.) sejam utilizados com **eficiência**, resolvendo os conflitos gerados pela competição pelos recursos, o Sistema Operacional deve:
 - Acompanhar a **situação** da utilização dos recursos
 - **Determinar** qual **processo** deve receber qual recurso, em que **momento** e em qual **quantidade**
 - **Alocar e recobrar** recursos nos momentos convenientes

Administração de Memória

- Mantém o Sistema Operacional informado de quais partes da **memória** estão **em uso**, o tipo de uso, a quantidade de memória em uso em cada caso, bem como das demais **áreas de memória disponíveis**, não alocadas na ocasião.
- Mantém a informação de **quais programas estão utilizando** as partes alocadas.
- Decide **quais processos podem receber** as áreas de memória que solicitam, quando, e quanto.
- **Aloca** memória a quem a solicitou, se for julgado possível e conveniente.
- **Recobra** para o sistema as áreas de memória das quais os processos não precisam mais, ou na ocasião em que completarem as suas atividades.

Modos de administração de memória

- São consideradas várias possibilidades para a **administração de memória**, escolhidas em cada caso conforme a realidade do sistema:
 - **Contígua** simples
 - **Particionada**, simples e relocável
 - **Paginada**, física e virtual
 - **Segmentada**, física e virtual
 - **Segmentada com paginação**, física e virtual

Administração de Processador

- Informa o Sistema Operacional sobre os **processadores existentes** no sistema e seus **estados** (*Traffic Controller*).
- Decide **qual job deve usar** os processadores (*Job Scheduler*).
- Decide **qual processo usa** o processador, quando, quanto (*Processor Scheduler*).
- **Aloca processadores** quando necessário (*dispatcher*).
- **Recobra o processador** quando o processo termina, excede o tempo, abandona o processador, ou quando ocorre um erro.
- **Política** comumente usada no *processor scheduler*: filas de **prioridades**: Processos que têm prioridades iguais são atendidos na ordem de chegada. Processos mais prioritários "passam na frente".
- **Outros motivos** para o administrador de processador **recobrar o processador**: execução de uma chamada de supervisor, por exemplo para solicitar Entrada/Saída; ocorrência de overflow em operação aritmética; solicitação feita por um processo com prioridade mais alta.

Multiprogramação

- Administradores de memória que permitem distribuir partes da memória a diversos programas certamente têm necessidade de sistema operacional multiprogramado, que permite que **vários programas** convivam na memória, **revezando-se** no uso do **processador**, de acordo com a disponibilidade e com a necessidade de cada programa em cada momento.
- Uma **fila circular** de programas aguardando alocação de processador é utilizada pelo S.O. para coordenar a execução de um conjunto de programas em regime multiprogramado.
- Convém distinguir muito bem a **multiprogramação do multiprocessamento**: o último envolve vários processadores, enquanto o primeiro, apenas um.

Administração de Dispositivos

- **Informa** o Sistema Operacional sobre os dispositivos, canais, unidades de controle **disponíveis** no sistema (*I/O traffic controller*).
- **Aloca** os dispositivos da forma mais eficiente possível, tanto para o programa como para o sistema (*I/O Scheduler*).
- **Recobra** para o sistema os dispositivos alocados ao programa, tão prontamente quanto for possível.

Classificação dos dispositivos

- Para efeito da aplicação adequada de **políticas de alocação**, os dispositivos de entrada/saída podem ser classificados, quanto ao tipo de acesso que lhes é inerente, em:
 - **Dedicados**: são alocados a um único processo de cada vez (leitora, perfuradora).
 - **Compartilhados**: (disco, fita) são "simultaneamente" utilizados por vários processos.
 - **Virtuais**: não existem fisicamente, apenas logicamente.
- É função do administrador de dispositivos promover a **distribuição** dos dispositivos disponíveis entre os **processos**, classificando-os em uma das categorias acima, de acordo com a conveniência, para melhor desempenho do sistema.

Administração de Informação

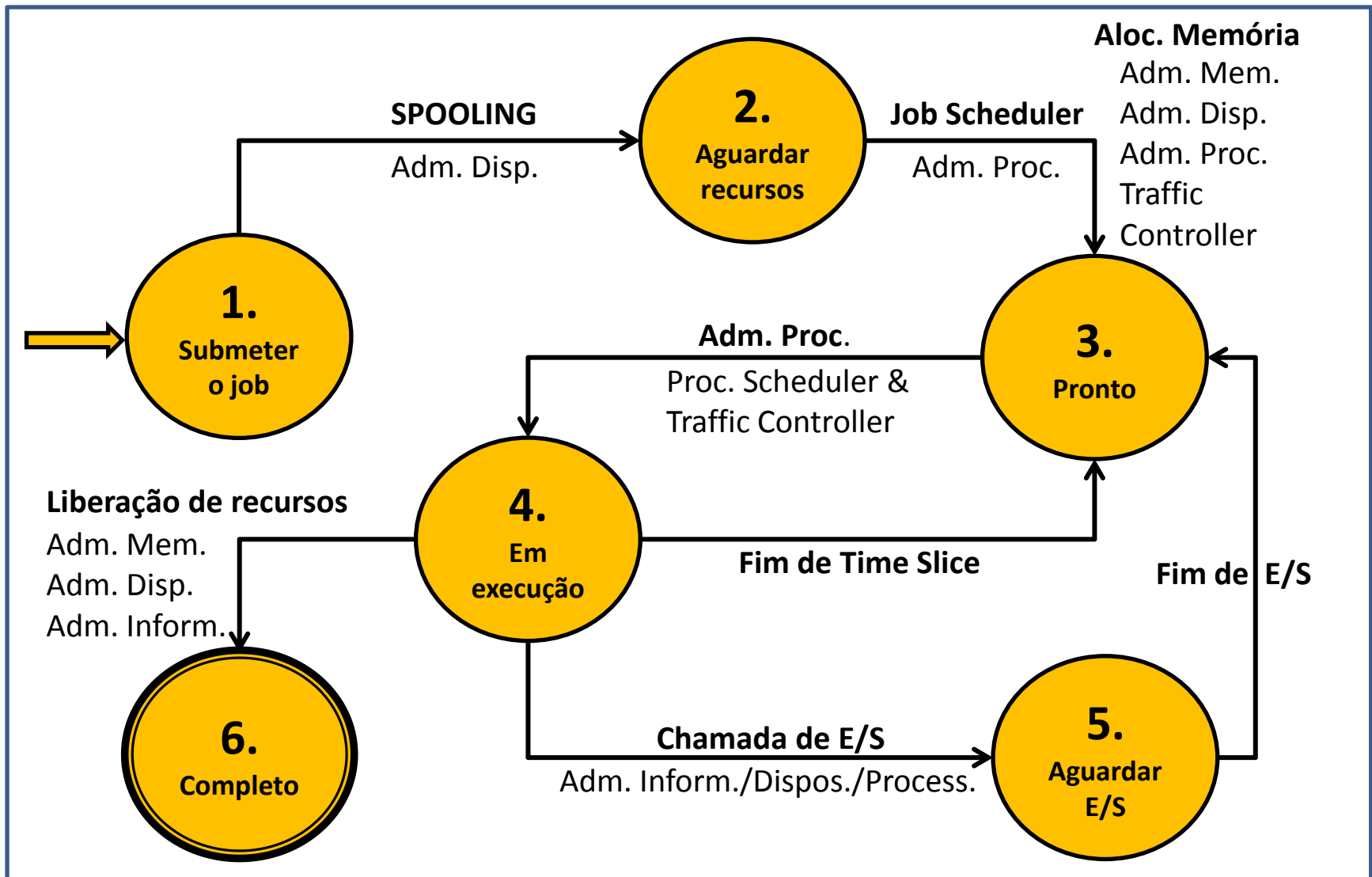
- **Informa** o Sistema Operacional sobre a localização, o uso e o **estado** das informações (especialmente arquivos de programas e dados) presentes no sistema (***File System***).
- Decide quem usará as informações, garantindo que certos dados e programas tenham como usuários apenas os processos que os criaram, ou outros explicitamente autorizados por aqueles (**proteção de informação**).
- **Recobra** para o sistema os arquivos já utilizados (*close*) bem como os recursos não mais necessários ao programa.
- Os **arquivos**, que podem ser públicos ou particulares, podem permitir acessos: de somente **leitura**, de **leitura e gravação**, ou de **execução**.
- O administrador de informações manuseia ainda rotinas do sistema, programas e bancos de dados, e permite o intercâmbio de informações entre os usuários e o sistema, e entre um usuário e outro.

S.O. COMO COORDENADOR DOS ESTADOS DO SISTEMA

Coordenação dos estados dos processos

- O Sistema Operacional pode ser visto como um **Coordenador das Mudanças de Estado** dos Processos.
- Entrando no sistema, cada programa passa por uma série de estados, conforme evolui sua execução.
- O software que se encarrega de **coordenar as transições de estados** dos processos em atividade é o Sistema Operacional.
- Os **eventos** que provocam a execução das atividades do sistema operacional responsáveis pelas mudanças de estado dos processos são as **interrupções**, que constituem **eventos assíncronos**, salvo em raras situações, como é o caso das interrupções solicitadas ou provocadas pelo próprio programa em execução.

Modelo de estados



Transições

1 → 2 Adm. Dispositivos: spooling

- Adm. Processador (Job Scheduler)

2 → 3 Memória disponível e Alocação

- Adm. Memória
- Adm. Dispositivos
- Adm. Processador
- Traffic Controller

Adm. Processador (*Processor Scheduler*)

Adm. Processador (*Traffic controller*)

4 → 3 Esgotou o tempo (*Time slice*)

- Revezar processador (fim de time-slice)
- Saída Anormal (Excesso de Tempo do Job)

3 → 4 Recebe o processador para executar

- Primeiro na fila dos processos prontos e processador livre

4 → 5 Chamada de operação de entrada/saída

- Adm. de INFO (p/iniciar Entrada/Saída)
- Adm. Dispositivos (para criar processos Entrada/Saída)
- Adm. de Processador (p/bloquear processo)

5 → 3 Entrada/Saída completada

4 → 6 Liberação de Recursos

- Adm. Memória
- Adm. Dispositivos
- Adm. Informações

Cada Job passa por transições dentro de uma máquina de estados como a esboçada antes.

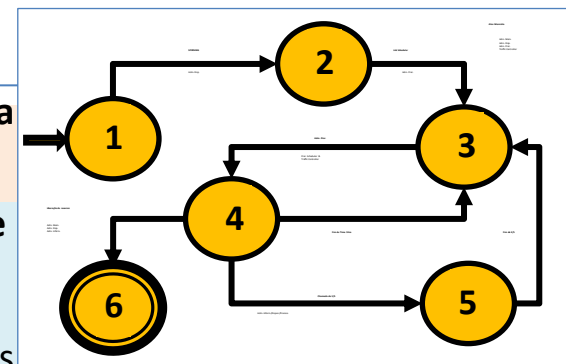
process scheduler

- seleciona um processo e aloca processador para ele
- efetua a transição 3 → 4.

traffic controller

- acompanha o estado dos processos
- fornece mecanismos para mudar o estado dos processos
- coordena a sincronização dos processos
- coordena a comunicação entre os processos

A seguir são detalhados os módulos básicos citados acima.



O S.O. COMO MÁQUINA ESTENDIDA

Máquina Hierárquica Estendida

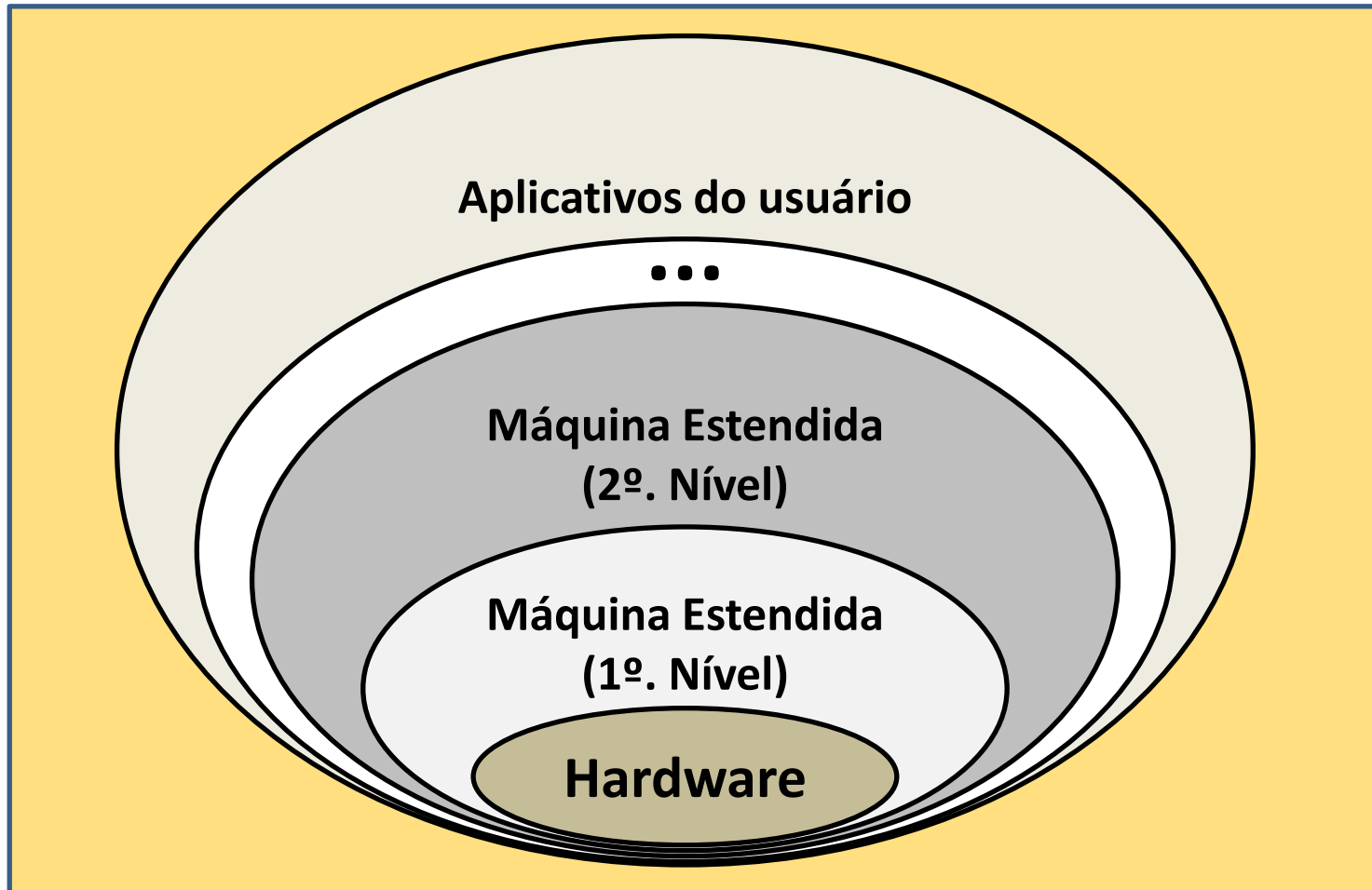
- Como foi dito anteriormente, o sistema operacional pode ser considerado uma espécie de prolongamento do hardware.
- **Máquina estendida** é uma abstração que corresponde ao conjunto das funcionalidades proporcionadas pelo hardware, acrescidas do conjunto das funcionalidades proporcionadas pelas chamadas de supervisor implementadas no sistema operacional.
- Do ponto de vista de implementação, caso o sistema operacional estiver projetado de tal modo que sua administração seja facilitada por meio de uma boa estruturação do programa, isso torna disponível uma máquina hierárquica.
- Nela, as funções mais essenciais e de nível mais baixo (mais próximo do hardware) são desempenhadas por uma máquina estendida situada em uma camada hierárquica mais interna.
- Dessa forma, apoiada diretamente no hardware, torna-se possível, com seu auxílio, a construção de programas da camada hierárquica imediatamente seguinte, e assim sucessivamente até a camada mais externa, em que estarão os aplicativos do usuário.

Máquinas Hierárquicas

- O conceito de **máquina estendida** muda a maneira de visualizar o processador, e o de **máquina hierárquica** torna a administração do computador mais adequada ao Sistema Operacional;
- Na máquina **hierárquica**, as funções mais básicas são alocadas nas máquinas estendidas posicionadas nas camadas mais internas;
- Em **cada camada** da hierarquia, as máquinas estendidas só têm acesso aos recursos fornecidos por camadas internas a ela;
- As máquinas **estendidas** que implementam abstrações complexas são posicionadas nas camadas hierárquicas mais externas.
- A estruturação, a **modularização e a hierarquização** das funcionalidades de programas de qualquer natureza (mesmo que não sejam sistemas operacionais) são práticas favoráveis à depuração dos mesmos, auxiliando o programador a conceber, documentar, implementar, depurar os seus programas, bem como colocá-los rapidamente em funcionamento confiável.

Organização hierárquica

- O esquema conceitual abaixo mostra a estruturação de programas e sistemas organizados como máquinas hierárquicas.



PRESTAÇÃO DE SERVIÇOS

Serviços

- Do ponto de vista dos sistemas de programação, o sistema operacional proporciona e implementa muitos **serviços** importantes, entre os quais:
 - operações de **entrada/saída**
 - tratamento de **interrupções**
 - manuseio de **recursos privilegiados**
 - tratamento das **chamadas de supervisor**
 - protocolos de **comunicação**
 - primitivas de **sincronização, paralelismo e concorrência**

INTERFACE COM O USUÁRIO

Comunicação com o usuário

- Uma importante contribuição dos sistemas operacionais ao conforto de seus usuários é a implementação de **formas de comunicação** com o usuário, para simplificar a operação do sistema.
- Destacam-se, dentre as **interfaces com o usuário** mais utilizadas :
 - Linguagens **textuais** (interpretação de linhas de comando)
 - Interfaces **gráficas** (comandos via mouse e tela gráfica)
 - Comandos de **voz** (interpretação de ordens vocais)
 - Comandos **gestuais** (via sensores de movimento)

ATENDIMENTO DE CHAMADAS DE SISTEMA

Chamadas de sistema

- Funções importantes implementadas pelo sistema operacional são usualmente implementadas na forma de **chamadas de sistema** (ou **chamadas de supervisor**).
- Isso é feito associando-se códigos aos serviços correspondentes, e ativando esses serviços mediante a execução da instrução especial de chamada de supervisor, a qual gera um **pedido de interrupção** ao sistema operacional.
- O atendimento dessas interrupções inclui a identificação do código fornecido pelo programa através da instrução de chamada de sistema e o acionamento de uma **rotina de tratamento** correspondente.
- Muitos sistemas operacionais disponibilizam os mesmos serviços por **outras vias**, tais como **comandos textuais** ou de **rotinas e aplicativos da biblioteca** de sistema.

COMUNICAÇÃO E SINCRONIZAÇÃO DOS PROCESSOS

Competições entre Processos

- Surge da necessidade de **compartilhar recursos** do sistema entre os processos.
- Devido ao compartilhamento de recursos, surgem **condições de competição** e os *deadlocks*.
- Ocorrem **condições de competição** sempre que vários processos requisitam simultaneamente os mesmos recursos.
- Nestas condições, é possível o aparecimento de **competições entre os processos** pelo uso de algum recurso compartilhado (por exemplo, uma impressora, área de memória, etc.)
- Nestes casos, dependendo da **ordem de alocação**, os resultados globais do processamento podem ser diferentes.

Disciplina de acesso aos recursos

- Por **exemplo**, as saídas geradas por uma **impressora** podem ser totalmente diferentes, dependendo da ordem em que ela é escalada para atender os processos requisitantes.
- Se não se tomarem os devidos **cuidados na alocação** da impressora aos diversos processos, as listagens poderão, inclusive, sair totalmente embaralhadas.
- Para este último problema, uma possível solução é a de exigir que o processo sempre **requeira o uso** de um recurso antes de poder efetivamente utilizá-lo.
- O recurso passa a estar **bloqueado para outros** processos até que seja liberado.

Sincronização de Processos

- Além dos dispositivos, também as **tabelas e arquivos** compartilhados **necessitam**, para seu uso, de operações de **sincronização**.
- Por exemplo, quando o recurso é a "**ready-list**", e o processo é o "**scheduler**", que é um programa que determina qual deve ser o próximo processo a receber o processador.
- Se houver **mais de um processador** no sistema, poderá haver problema de **conflito**: possivelmente em um dado instante, dois processadores estarão disponíveis para executar o mesmo processo, e possivelmente algum processo poderá deixar de ser executado devido à aplicação de um **critério inconveniente** de alocação de processador.

Cooperação entre processos

- Uma solução para este problema é a utilização de um "**lock byte**", variável que indique se os dados estão ou não em uso no instante da requisição.
- Observe-se que é fundamental que haja **cooperação** entre os processos, sem o que o sincronismo não funcionará.
- Se um processo, proposital ou acidentalmente, **deixar de liberar seus recursos**, tais recursos permanecerão **inacessíveis** aos demais processos do sistema.
- Por isso, por precaução, os sistemas operacionais costumam **liberar todos os recursos** do processo ao ocorrer o seu **final de execução**, para garantir a recuperação, pelo sistema, dos recursos a ele alocados, mas não espontaneamente liberados.

Operações Primitivas

- Note-se que, **uma vez iniciados**, todos esses procedimentos devem ser **executados até o final** sem interrupções, ou seja, tais operações devem ser **monolíticas**, implementadas como regiões críticas, caso contrário, haverá o risco concreto de inconsistências na sincronização.
- A exemplo de outras situações envolvendo a atualização de informação do sistema, essas operações devem ser sempre realizadas, estando **desativado o sistema de interrupção** da máquina.
- Rotinas do sistema com essas características são chamadas **operações primitivas**, ou operações **atômicas**, pois se comportam **monoliticamente**, como se fossem instruções de máquina: uma vez iniciadas, só poderá haver o atendimento de pedidos de interrupção após o término de sua execução.

Primitivas p/ sincronização e comunicação

Todas essas primitivas são executadas com a interrupção inibida.

- **LOCK/UNLOCK** – Mecanismo primitivo de sincronização que fecha/abre o acesso a uma região crítica do programa.
- **WAIT/SIGNAL** – Mecanismo primitivo para efetuar o bloqueio/desbloqueio de processos.
- **P/V** – Mecanismo de utilização de semáforos contadores para sincronizar processos solicitando/liberando recursos compartilhados.
- **SEND/RECEIVE** – Mecanismo de sincronização usando o sistema de comunicação para enviar/aguardar o recebimento de mensagens que tenham a finalidade de comunicar e/ou sincronizar os processos mutuamente.

**MODERAÇÃO DE POLÍTICAS,
DIREITOS E PRIORIDADES**

Prioridades

- A política de inserção, por prioridade, de um programa na fila de processamento resulta da **ordenação** dessa fila **de acordo com a prioridade** atribuída a esse programa.
- O resultado da aplicação desta **política** depende, portanto, do **critério** (muitas vezes, arbitrário) adotado para a **atribuição de prioridades** aos processos.

Balanceamento do Sistema

- Esta política dá **preferência a processos** que efetuam **muitas operações de Entrada/Saída**, como é o caso de softwares comerciais, com a finalidade de procurar, na medida do possível, manter ocupados os equipamentos de entrada/saída, **evitando assim a ociosidade dos periféricos.**

Mínimo Tempo de Resposta

- Esta política dá preferência a **processos interativos**, para que seja obtida uma **rapidez maior no atendimento a terminais *on line***.
- Isto se justifica pelo efeito psicológico a que está sujeito um **operador** que utiliza o sistema através de um terminal: ele sente-se em geral **desconfortável** quando tem a sensação, mesmo que falsa, de que o sistema operacional **não esteja dando permanente atenção** a seu terminal.

Prioridade dinâmica

- Às vezes é conveniente que o próprio sistema estabeleça (**automaticamente**) **prioridades** para os processos, durante a sua execução, de acordo com os **méritos e deméritos dinâmicos** dos mesmos, ou então buscando **balancear a carga** do computador, ou ainda para atingir algum **índice de mérito** estabelecido como meta.

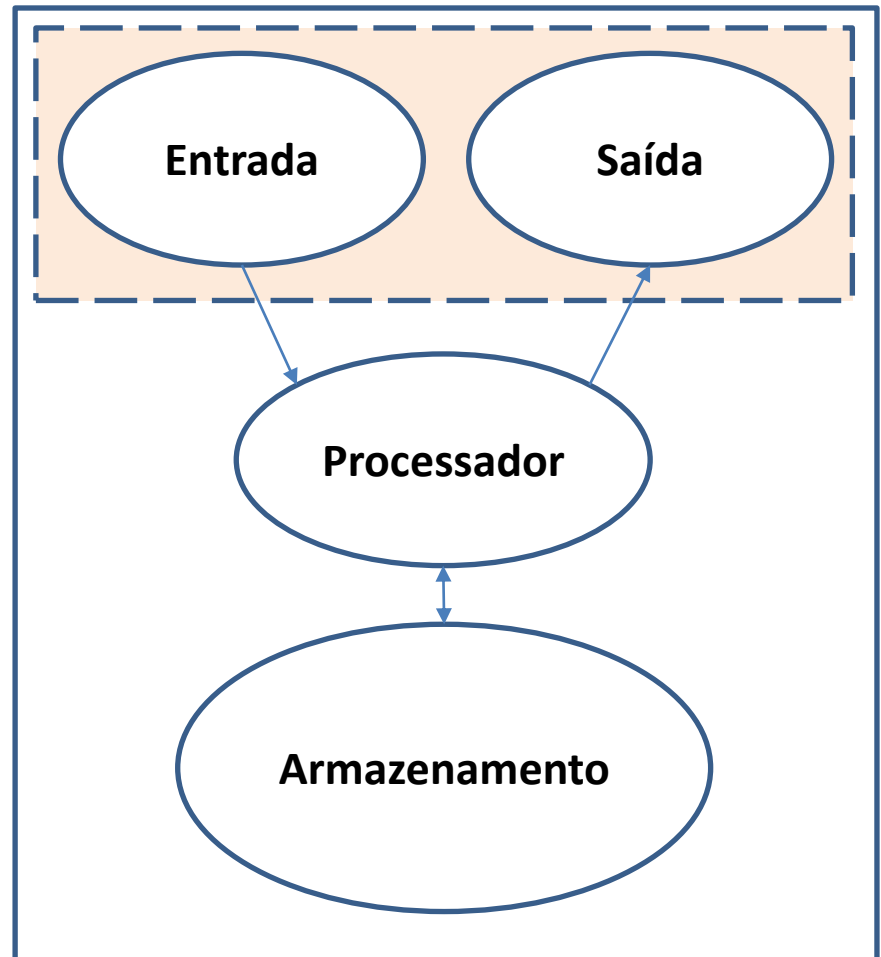
Política de alteração de prioridade

- Às vezes o sistema **umenta a prioridade** de um processo que **utiliza muitos recursos**, pois dessa forma ele pode ser favorecido para terminar mais depressa, e assim liberar os muitos recursos a ele alocados, o que não seria possível comodamente se a sua **prioridade** não pudesse ser **alterada dinamicamente**.

SERVIDOR DE ENTRADA/SAÍDA

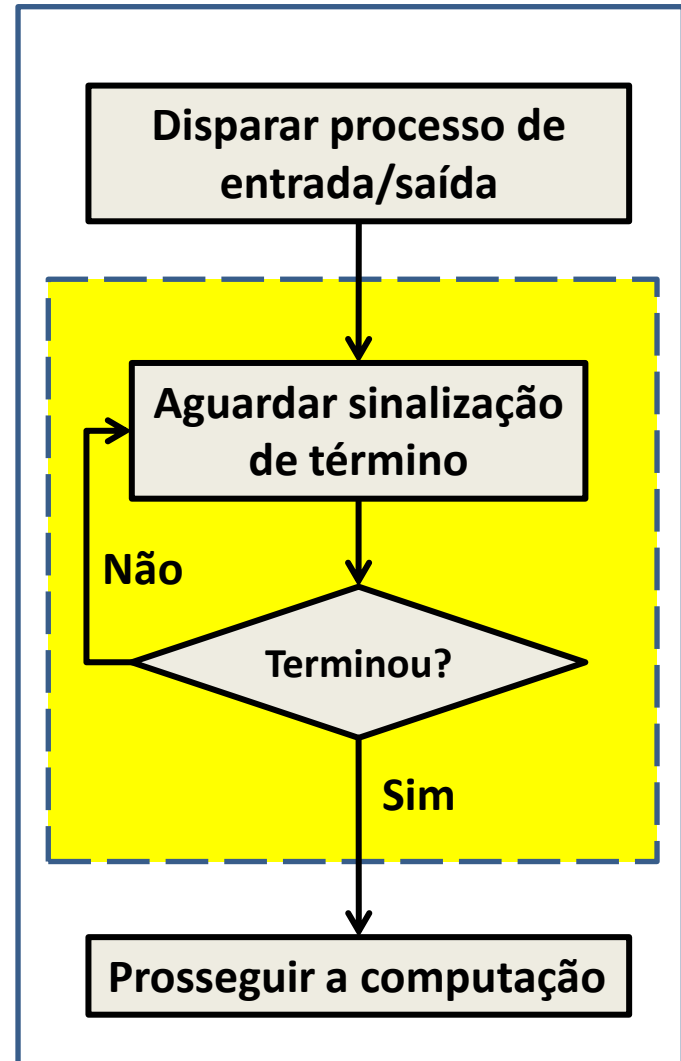
Entrada e saída

- Toda ação ocorrida em um sistema de computação que **não seja processamento nem movimentação de dados na memória principal** pode ser enquadrado na classe de operação de entrada e saída.



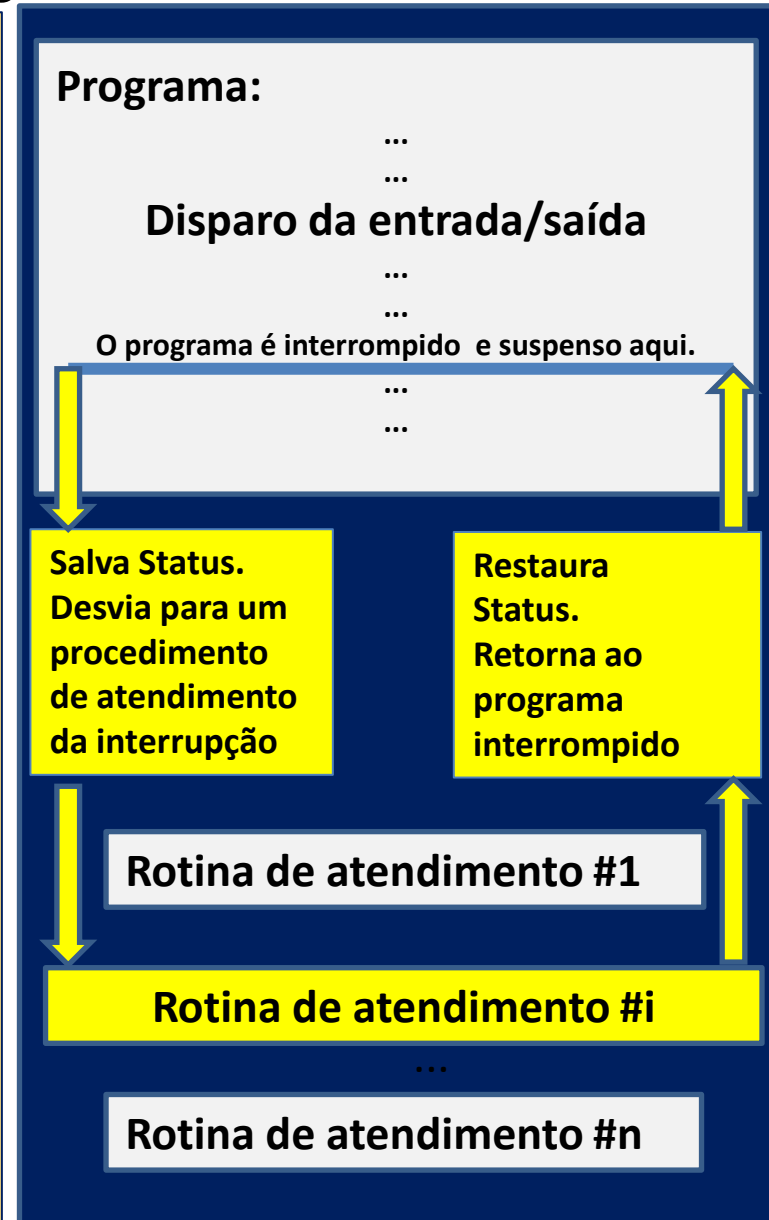
Espera ocupada

- A forma mais trivial de sincronização entre o programa e a entrada/saída realizada consiste na **espera ocupada**.
- Neste método, o programa explicitamente se mantém todo o tempo perguntando se a operação de entrada/saída solicitada já foi concluída.
- Apesar de ser uma técnica muito simples e de fácil implementação, tem a desvantagem de utilizar o **processador durante todo o tempo** sem produzir qualquer resultado que seja útil para o programa.



Interrupção

- Isso motivou a invenção da técnica da *interrupção*, em que o hardware e o programa operam *em paralelo*.
- O *Hardware sinaliza* o momento do *término da operação em andamento* (ou seja, *gera um pedido de interrupção*)
- O programa é *suspenso*, seu *status é salvo* e uma *rotina de tratamento* é executada.
- Ao fim do tratamento, o *status é restaurado* e o programa continua normalmente.
- *Pedidos de interrupção diferentes* podem ser distinguidos por hardware, e assim receber *tratamentos distintos*.
- Uma boa vantagem deste método é que a *CPU não é ocupada* nessa tarefa, nitidamente administrativa.



TRATAMENTO DE EXCEÇÕES E INTERRUPÇÕES

Rotinas de tratamento de interrupção

- A estrutura típica de uma **rotina de tratamento de interrupção** envolve:
 - **Salvar o status** do programa interrompido
 - Desviar para uma **sub-rotina específica**, para tratar o evento ocorrido, empilhando o endereço de retorno (finalizar a Entrada/Saída em andamento e iniciar a próxima, se houver)
 - **Sinalizar** por software, se for o caso, **o final do atendimento** ao pedido do usuário
 - **Restaurar o status** e retornar.
- Este foi o mecanismo que inspirou a arquitetura dos **motores de eventos**, anteriormente estudados.

Tipos de interrupção

- Há diversos **tipos de interrupção** que costumam ser encontrados na maioria dos sistemas computacionais, tais como:
 - De **entrada/saída** (geradas por eventos de final de operações de entrada/saída e comunicação)
 - De **relógio** (geradas pelo transcurso do tempo)
 - De **falha de hardware** (gerada internamente por falhas detectadas pelos circuitos da máquina)
 - Do **operador** (solicitada pelo operador da máquina por pressionamento de botões de controle do computador)
 - De **software** (explicitamente solicitada por um programa através da instrução de chamada de sistema)

VIRTUALIZAÇÃO DE RECURSOS

Virtualização de Memória

- Implementa técnicas de *virtualização de memória* utilizando dispositivos de alta capacidade de armazenamento, tais como discos, para simular áreas de memória não existentes, de forma que ao usuário seja produzida a **ilusão da disponibilidade** de muito mais memória do que existe na realidade.
- Com a ajuda de **recursos de hardware**, a utilização da memória virtual tem um **baixo impacto no desempenho** da máquina e dos programas, de modo que tudo se passa como se a máquina realmente tivesse memória física muito grande.

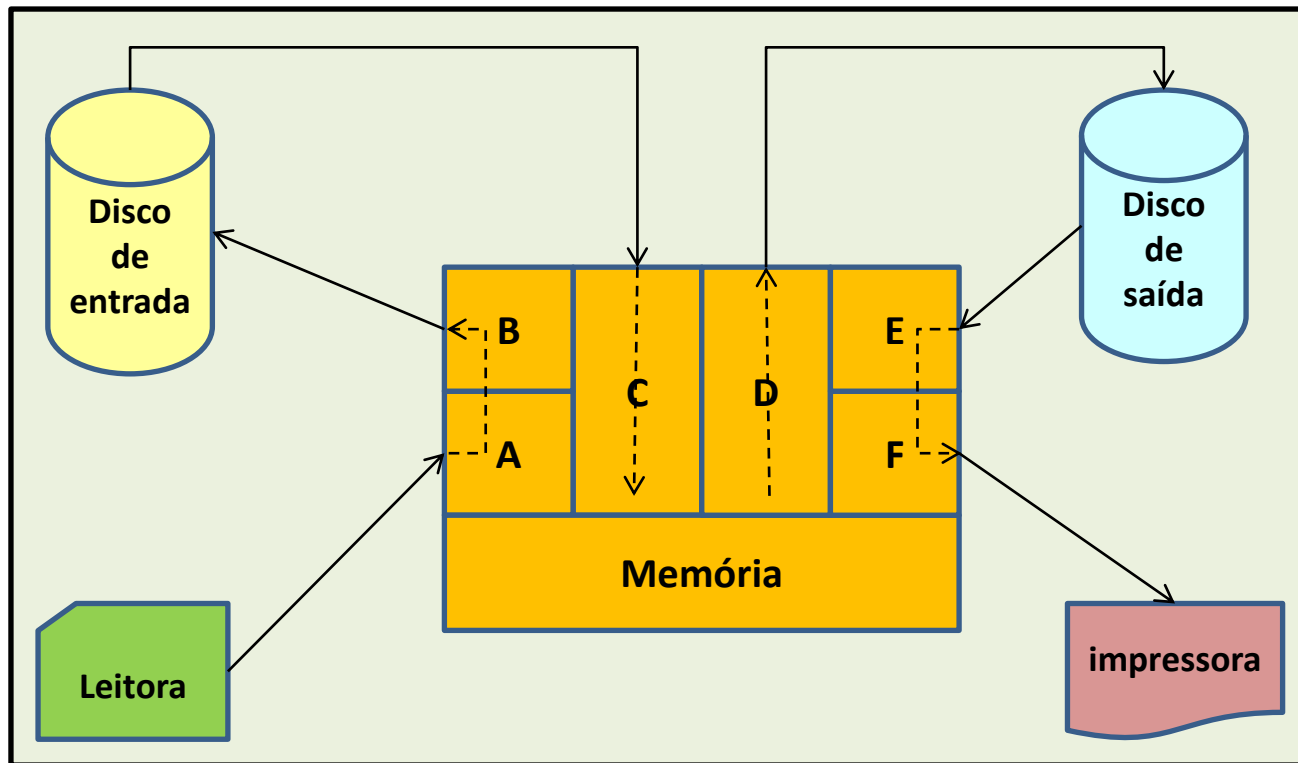
Virtualização de Processador

- Este assunto já foi estudado nesta disciplina, com o nome de *simulação de máquinas virtuais*.
- A técnica da *virtualização* de processadores permite ao usuário utilizar localmente programas desenvolvidos para outras máquinas sem que haja necessidade de migrações físicas de seus programas.

Virtualização de Dispositivos

- Outra função do Administrador de Dispositivos é o de executar "***SPOOLing***", que consiste em copiar, em dispositivos de armazenamento de massa de grande capacidade e de acesso rápido, imagens das entradas e saídas (lentas) desejadas.
- Com isso, criam-se ***dispositivos virtuais*** pela sua simulação em arquivos de um disco ou outro armazenamento rápido e de alta capacidade, melhorando globalmente a eficiência do sistema.
- **Processadores periféricos** podem efetuar o trabalho de mover os dados do dispositivo para o disco ou vice-versa enquanto o processador principal executa outras operações.
- Através da ***virtualização de periféricos***, o ***SPOOLing*** facilita o programador em tarefas que necessitam de mais periféricos que os fisicamente existentes, permitindo assim que o usuário utilize ***periféricos lógicos***, os quais são **simulados** em arquivos do disco, os quais, para todos os efeitos, **comportam-se**, do ponto de vista dos seus usuários, **como se fossem dispositivos físicos reais**.

Esquema de um sistema de E/S virtual



Programas de canal:

(A) - entrada pela **Leitora**

(B) - saída para **disco de Entrada**

(C) - entrada do **disco de Entrada**

(D) - saída para **disco de Saída**

(E) - entrada do **disco de Saída**

(F) - saída para a **Impressora**

SUORTE À SEGURANÇA NO SISTEMA

Principais técnicas de segurança no sistema computacional

- Para impedir, ou ao menos reduzir os danos provocados pela quebra de segurança em um sistema computacional, têm sido adotadas algumas **técnicas**, dentre as quais as mais importantes, e que serão comentadas aqui, são:
 - **autenticação**
 - **controle de acesso**
 - **detecção de intrusos**
 - **defesa contra *malware***
 - **cuidados com ataques de *buffer overflow***

Autenticação

- A **verificação de identidade** é usada para **garantir que a entidade esteja autorizada** a usar os recursos do sistema, e para isso considera dois passos:
 - Passo de **identificação** – a entidade apresenta um “*passaporte*” ao sistema de segurança.
Identificar uma entidade como válida é base para o esquema de segurança e abre portas para os demais serviços do sistema, logo requer muito cuidado.
 - Passo de **verificação** – ocorre quando da autenticação do “passaporte” apresentado, procurando assim **associar** de forma segura a **entidade** com a sua **identificação**.

Controle de acesso

- Estabelece os tipos, circunstâncias e agentes dos acessos permitidos, e pode ser efetuado de três modos (eventualmente aplicados em conjunto):
 - **Discricionário** – Acessos aos recursos do sistema são concedidos ou não por decisão unilateral do sistema de controle.
 - **Mandatário** – Acessos são concedidos só se o nível de segurança do requisitante for compatível com as permissões de acesso ao recurso, estabelecidas para o requisitante.
 - **Baseado no papel da entidade** – Baseia-se na compatibilidade entre a função desempenhada no sistema pelo requisitante e regras preestabelecidas para o acesso.
- **Matrizes e listas de controle de acesso** costumam ser utilizadas para a implementação desta técnica, e relacionam as informações sobre os direitos do requisitante com as restrições que o sistema impõe quanto ao acesso dos usuários ao recurso solicitado.

Detecção de intrusos

- Para a presente discussão, são relevantes dois conceitos:
 - **Intrusão na segurança** – evento ou combinação de eventos que caracteriza um incidente de segurança de acesso ao sistema, ocorrido sem autorização.
 - **Detecção da intrusão** – monitorando e analisando os eventos do sistema, um serviço localiza e denuncia tentativas ou ocorrências de acessos não autorizados ao sistema.

Requisitos

- Este grupo de técnicas se concentra em **determinar a presença de intrusos** e o seu efeito no sistema. Para isso, deve contar com:
 - **Sensores**, responsáveis pela coleta de dados, compreendendo pacotes de rede, arquivos de log, relações de chamadas de sistema.
 - Os sensores enviam os dados coletados aos **analísadores**, que também podem receber dados provenientes de outros analisadores, indicando se foi detectada a presença de algum intruso no sistema.
 - Como saída, os analisadores enviam a uma **interface com o usuário** o diagnóstico realizado, e as evidências encontradas que levaram a tal diagnóstico.

Técnicas de detecção usuais

- As técnicas usuais de detecção de intrusos abrangem
 - detecção de **anomalias de comportamento**
 - detecção de **assinaturas de intrusos**.
- Ferramentas essenciais para esta família de técnicas de segurança são os **registros de auditoria**, que podem ser
 - ***Nativos***
(coletam informação sobre a atividade normal do sistema)
 - ***Específicos*** (exclusivos para alguma particular ocorrência)e indicam:
 - *o sujeito, a ação, o objeto*
 - *a condição de exceção, o recurso afetado e o uso dele efetuado*
 - *uma identificação temporal única da ocorrência*

Defesas contra *malware*

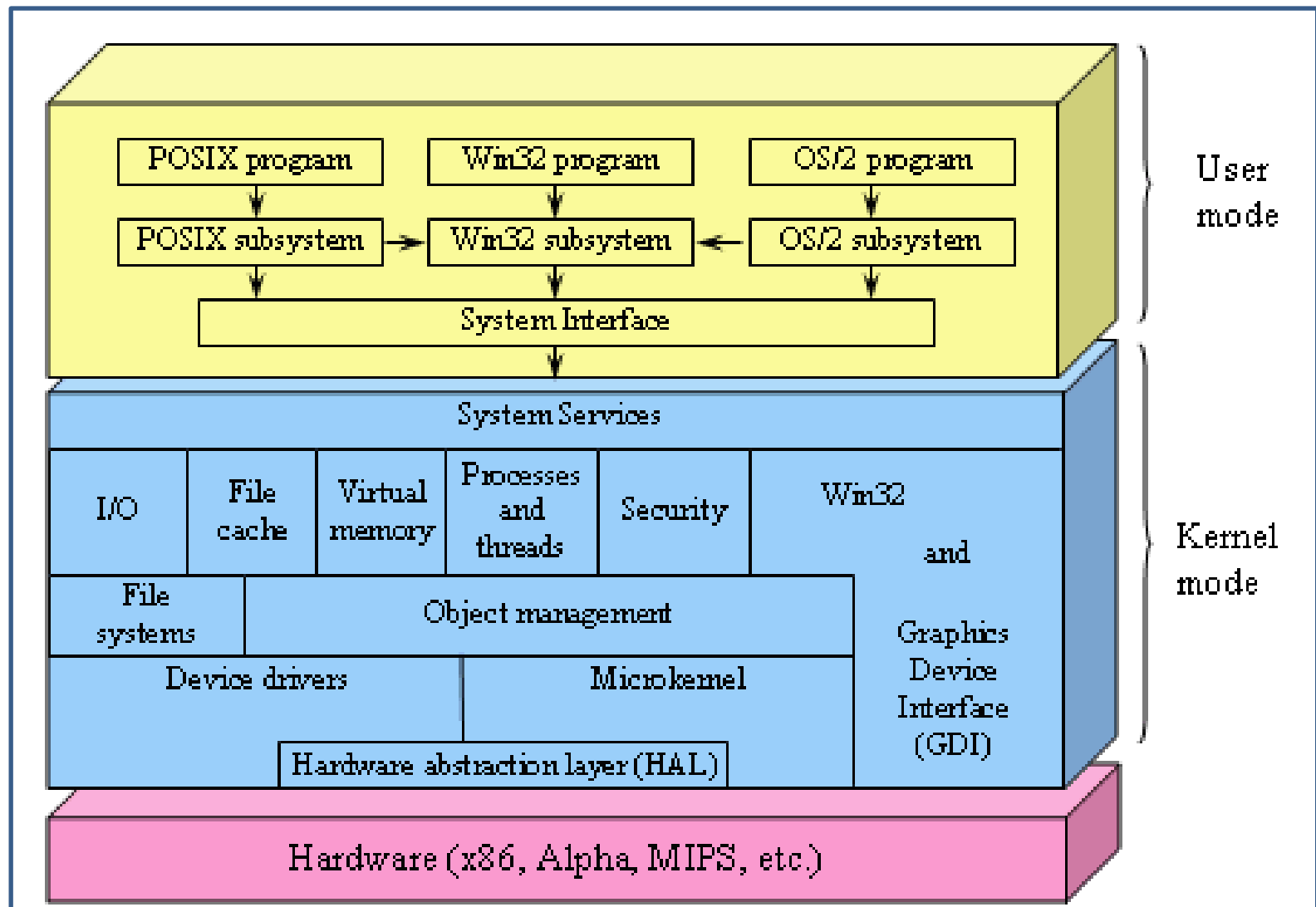
- Nesta classe incluem-se as medidas profiláticas contra intrusões. **Antivírus** envolvem:
 - **Detecção** – uma vez infectado, *identificar* essa situação e *localizar* sua causa.
 - **Identificação** – após constatar sua existência, *descobrir a praga atuante específica*
 - **Remoção** – *restaurar o sistema* a seu estado original pela *destruição* do fator contaminante
 - **Reinstalação** – Detectando a praga, mas falhando nas outras ações, é instalada uma cópia não contaminada do programa.
 - **Ações específicas** – Particulares, para cada tipo de praga (vírus, vermes, *bots*, etc), sempre com a finalidade profilática de *neutralizar sua ação* e, preferencialmente, de *erradicá-la*.

EXEMPLO DE UMA ARQUITETURA DE SISTEMA OPERACIONAL: WINDOWS NT

A arquitetura de um Sistema Operacional

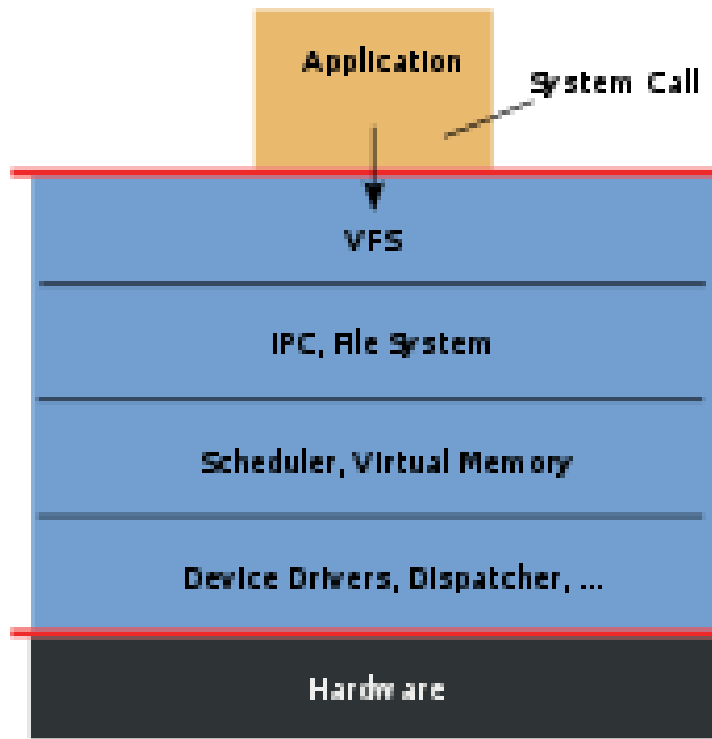
- Para ilustrar com um exemplo real de sistema operacional, escolhemos o **MS-Windows NT** para uma observação rápida da sua arquitetura, baseada no conceito de *microkernel*.
- Os dois slides a seguir mostram, respectivamente, o esboço da sua **arquitetura hierárquica** em camadas, e um comparativo entre a sua estrutura baseada em *microkernel* e estruturas clássicas simplesmente hierárquicas.

Arquitetura do *MS-Windows NT*



kernel monolítico x microkernel

Monolithic Kernel based Operating System

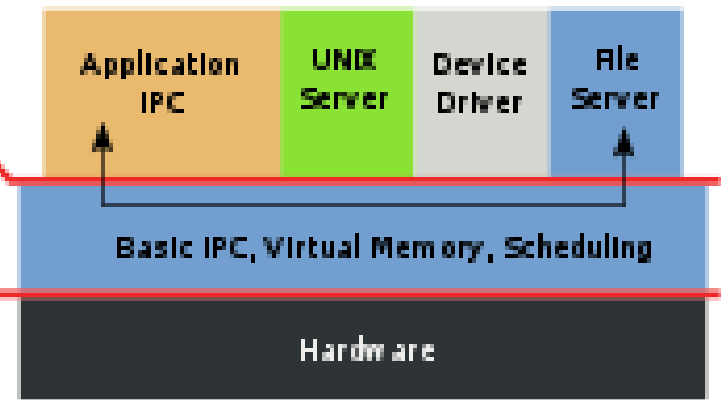


Microkernel based Operating System

user mode

kernel mode

VFS = virtual file system
IPC = Interprocess communication



Microkernel

- Como se pode observar, o uso de arquitetura baseada em *microkernel* remove do núcleo do sistema operacional todas as operações que não necessitem ser executadas estritamente em modo privilegiado.
- Isso **reduz** drasticamente o **processamento** feito em **modo supervisor**, liberando o hardware para continuar a executar outros programas em modo usuário.

CONCLUSÃO

Conclusão

- Com uma pequena **visita panorâmica** pretendeu-se aqui apresentar de forma qualitativa informações básicas sobre um conjunto significativo de tópicos estudados em obras sobre Sistemas Operacionais.
- Tais informações procuram proporcionar uma formação básica que situa o **sistema operacional** como um dos mais importantes componentes de um sistema de programação.
- Espera-se que esta aula tenha ajudado a completar a formação conceitual do aluno acerca de uma parcela significativa do software básico tipicamente encontrado nos sistemas computacionais modernos.

FIM