

SISTEMAS OPERATIVOS

CAPÍTULO 1. FUNDAMENTOS DE SISTEMAS OPERATIVOS MODERNOS

1. Generalidades sobre o sistema computacional
2. Visão abstracta de um Sistema Operativo
3. Evolução dos Sistemas Operativos. Tipos de Sistemas Operativos
4. Concorrência e Multiprogramação
5. Estrutura de um Sistema Operativo
6. Componentes e alternativas ao Sistema Operativo
7. Atributos de qualidade de um Sistema Operativo
8. Aquitectura e Classificação de Sistemas Operativos

1. Generalidades sobre o sistema computacional

Um computador de uso geral, moderno, consiste de uma CPU e em uma série de controladoras de dispositivos que são conectadas através de um barramento comum que fornece acesso à memória compartilhada. Cada controladora de dispositivo está encarregada de um tipo específico de dispositivo (por exemplo, unidades de disco, dispositivos de áudio e monitores de vídeo). A CPU e as controladoras de dispositivo podem executar em modo concorrente, competindo pelos ciclos de memória. Para garantir acesso correcto à memória compartilhada, uma controladora de memória é fornecida e sua função é sincronizar o seu acesso.

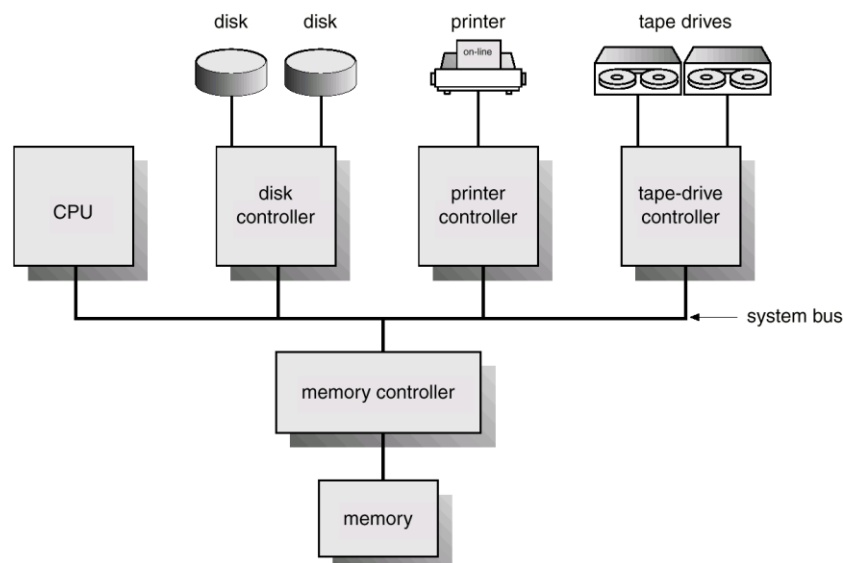


Fig.1 .. Sistema computacional moderno

2. Visão abstracta de um Sistema Operativo

Numa visão mais genérica, o computacional pode ser visto como um binómio: Hardware e Software. Este último é entendido como um conjunto de programas que comandam o funcionamento de um computador, isto é sequência de instruções escritas para serem interpretadas por um computador.

Um dos software mais importantes do sistema operativo moderno é o sistema operativo, que se enquadra na categoria do software do sistema, conforme a árvore que se segue.

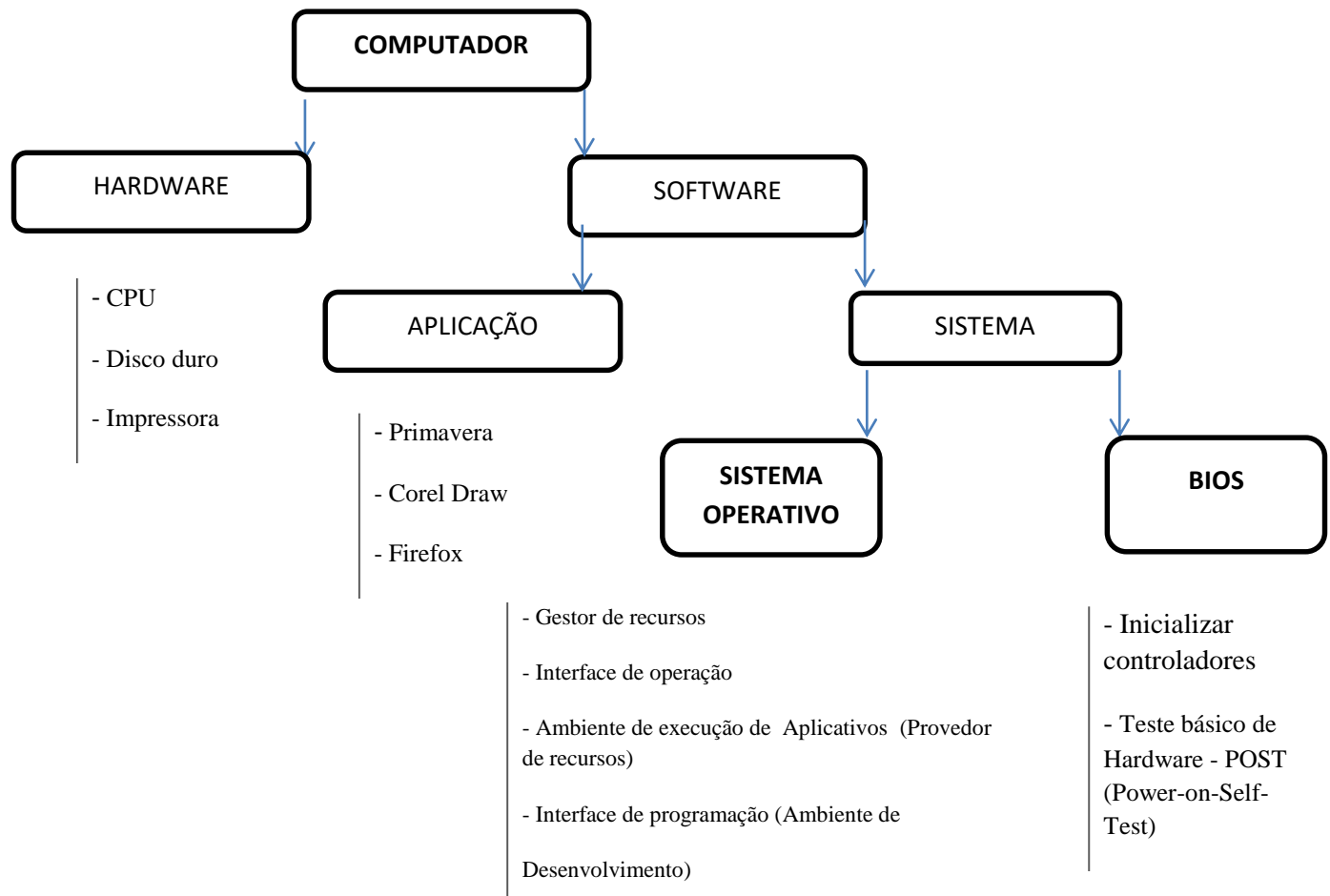


Fig.2 ∴. Visão do sistema computacional

Constituem objectivos de um Sistema Operativo:

- Compartilhar recursos de forma organizada
- Facilitar o acesso aos mesmo

Assim sendo, cumprindo com esses objectivos, o Sistema Operativo pode ser visto de diferentes formas:

- Utilizador – Interface de operação (UI - User Interface)
- Aplicação/compiladores - provedor de recursos
- Programadores/Analistas – Interface de programação, isto é o sistema operativo oferece todos recursos necessários para simplificar o desenvolvimento de aplicações.
- E numa visão de baixo para cima, o sistema operativo é tratado como gestor de recursos, que transforma máquina física em máquina virtual, que disponibiliza os recursos para as camadas superiores (aplicações, utilizadores, compiladores e programadores), sem descorar a a utilização eficiente dos mesmos (CPU, memória primária e dispositivos de E/S)

3. Evolução dos Sistemas Operativos. Tipos de Sistemas Operativos

Antes da década de 50, os computadores eram muito difíceis de serem programados. Era necessário conhecer totalmente sua arquitectura, e tal operação era efectuada em painéis com cerca de 6.000 conectores, em linguagem de máquina. Nesta fase os computadores não possuíam ainda dispositivos para interacção com o usuário.

Na década de 50, já com a utilização de transistores, sucedeu-se um grande avanço tecnológico, melhorando a velocidade dos processadores e a capacidade dos meios de armazenamento, em especial a memória e os discos magnéticos.

Por volta de 1953 foi introduzido o primeiro sistema operativo, um programa de controle que permitia uma interacção, mesmo que limitada, entre o operador e a máquina, optimizando a execução das tarefas. Em 1959 foi criada uma versão de sistema operativo que já implementava conceitos de memória virtual, conceito este largamente utilizado nos sistemas actuais.

Na década de 60, a partir do surgimento dos circuitos integrados, foi possível difundir o uso de sistemas computacionais em empresas, com diminuição de custos e tamanho dos equipamentos. Além disso, esta década presenciou inúmeras inovações na área de sistemas operativos, presentes até hoje, como os ambientes de multitarefa, multiprogramação, multiprocessamento e time-sharing, tendo o desenvolvimento destas técnicas avançado até o meado da década de 70, onde também foram

implementadas as tecnologias baseadas em arquitectura VLSI (chips), as primeiras redes de computadores, e o desenvolvimento de diversas linguagens de programação de alto nível.

A década de 80 foi marcada pela criação dos microcomputadores, baseados em microprocessadores de uso pessoal. Liderados pela IBM, diversos fabricantes seguiram por essa linha, porém alguns deles não abandonando a fabricação dos computadores de grande porte, como foi o caso da própria IBM.

Nota-se que, a partir do meado da década de 80, acontece uma divisão de águas, com a indústria passando a produzir equipamentos de grande porte e muitos modelos de microcomputadores, que também precisavam de sistemas operativos bastante evoluídos. Foram, então, utilizadas as técnicas modernas já existentes nos ambientes de grande porte na implementação de sistemas operativos para os microcomputadores, com versões diversas, todas inicialmente mono-usuário / monotarefa (devido à baixa capacidade de armazenamento dos computadores, naquela época). Com o avanço da tecnologia, os computadores passaram a integrar discos rígidos e outros periféricos, possibilitando a criação de sistemas operativos mais evoluídos nesta categoria de computadores, quando surgiram os sistemas mono-usuário / multitarefa, que são largamente utilizados até hoje.

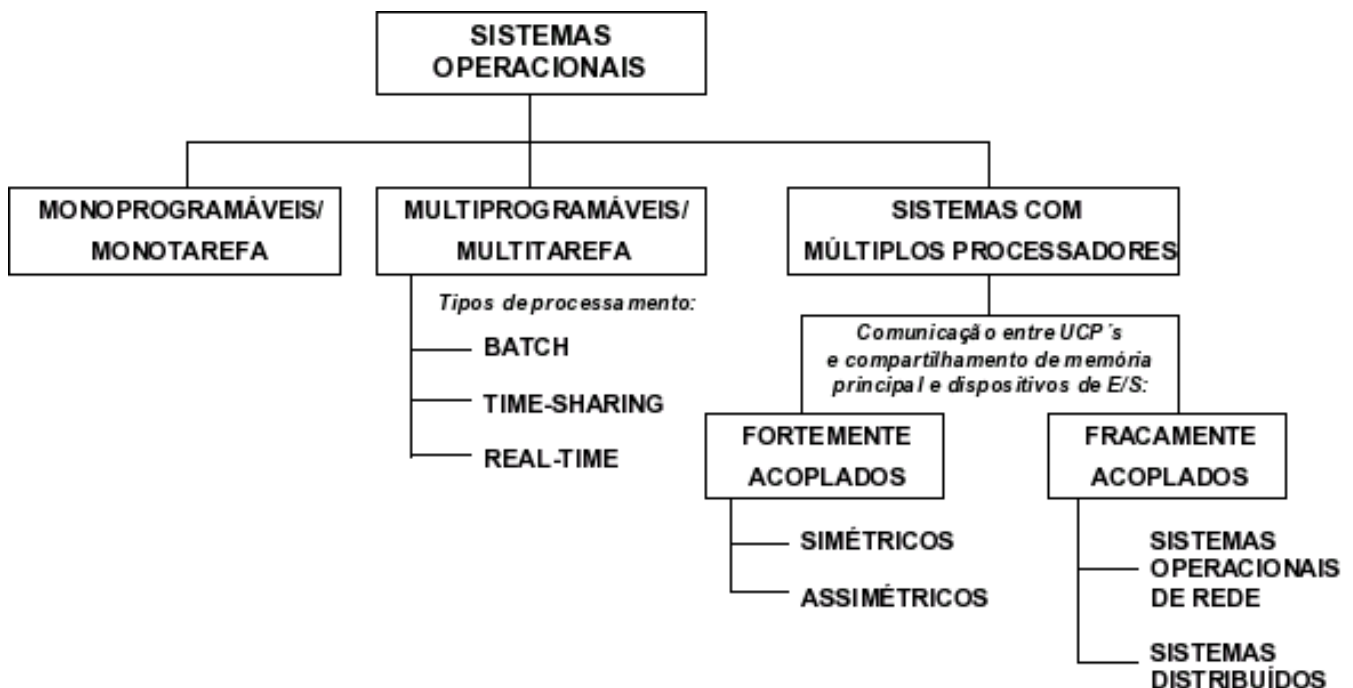
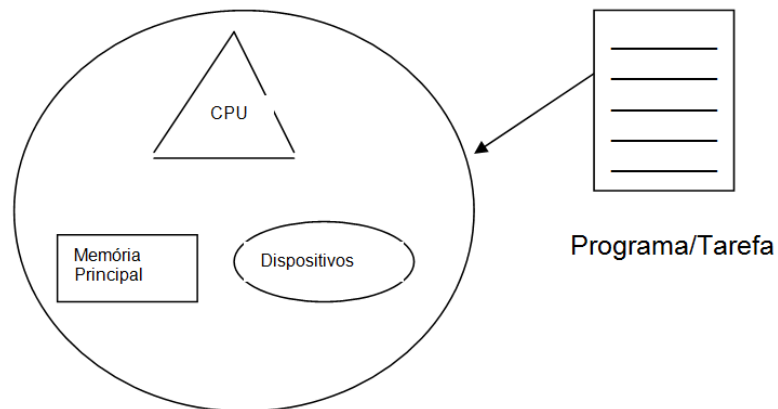


Fig.3 .. Tipos de sistemas operativos

Sistemas Monotarefa /multiprogramável

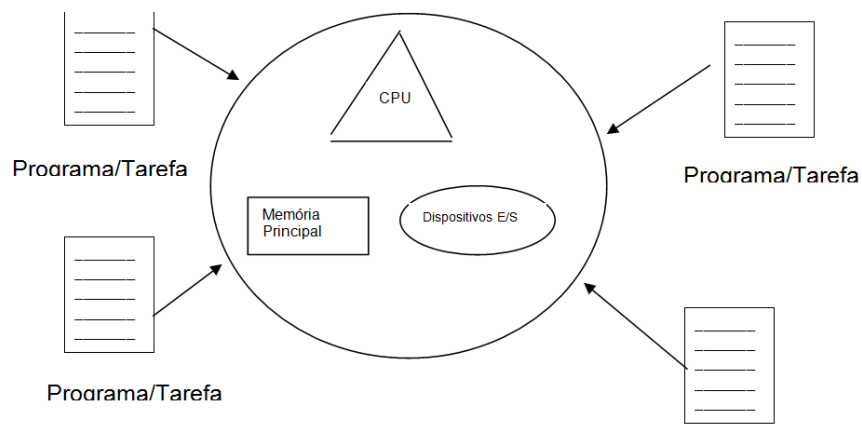
- Um único programa em execução.
- Todos os recursos dedicados a esse programa – desperdício de recursos
- Estrutura simples
Ex: MS-DOS, PALM OS



Sistema Monoprogramável/Monotarefa

Sistemas Multitarefa /multiprogramável

- Vários programas em execução.
- Compartilhamento e gestão dos recursos.
- Melhor aproveitamento.
- Estrutura complexa.
- Alguns são multi-usuários.



Sistema Multiprogramável/Multitarefa

Lote (Batch)

- Primeiros sistemas operativos multitarefa
- Tarefas não interactivas
- Fila de submissão de tarefas, sem interferência pois as tarefas não coexistem na memória
- Saída em impressora ou disco
- Podem ser eficientes, como também podem ter tempos de resposta longos
- Ideal para computação financeira e científica onde o mesmo tipo de processamento e análise são realizados com frequência; A transição entre as tarefas é automática.

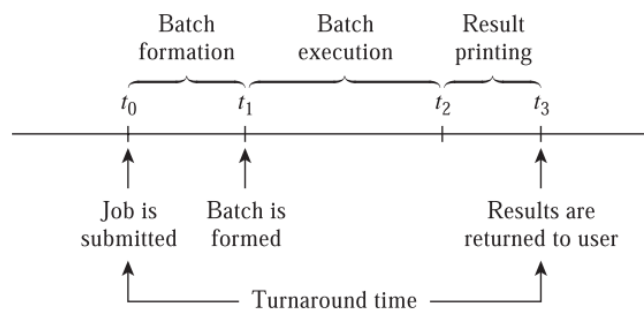


Fig .4 ∴ Processamento em lote

Tempo Compartilhado (Time-Sharing)

- Tipo mais comum
- Buscam eficiência na utilização de recursos: operações de I/O e processamento podem correr a qualquer altura, o que aumenta a performance;
- Conceito de fatia de tempo (*Time slice*)
- Cada tarefa fica em execução até o tempo atribuído, dando lugar a outra
- Tarefas interactivas com bons tempos de resposta, aplicando por exemplo o escalonamento Round Robin.

Ex: UNIX, WINDOWS

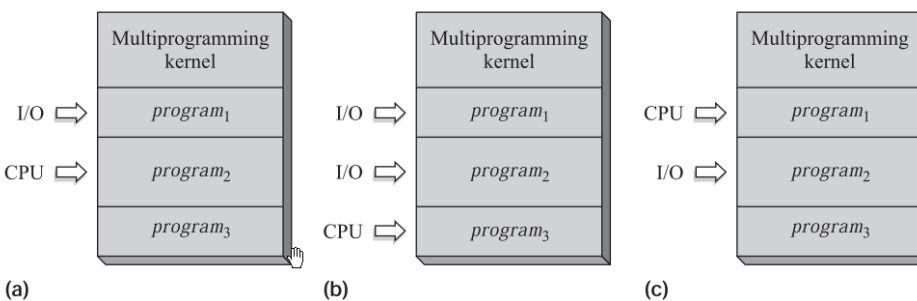


Fig .5 .: Sistemas de tempo compartilhado

Tempo Real (RTOS)

- Semelhante ao sistema time-sharing, mas não iguais.
- Tempo de resposta dentro de intervalos rígidos
- Conceito de prioridade: tarefa em execução enquanto não houver outro de mais prioridade do que ela
- Prioridade definida pela aplicação
- Uso em sistemas de missão crítica e controle de processos: refinarias, siderúrgicas, tráfego aéreo.

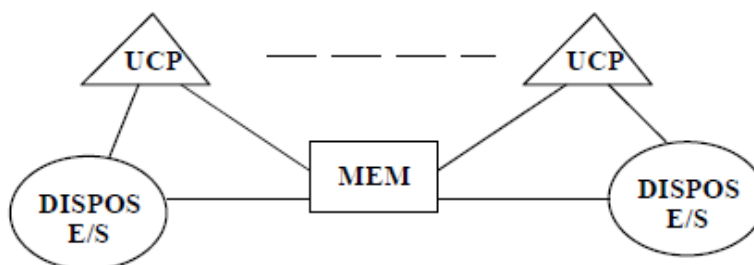
Múltiplas CPU's

- Processamento paralelo
 - Escalabilidade
 - Disponibilidade
 - Balanceamento de carga
 - Tendência actual
- Problemas:
 - Necessidade de desempenho
 - Mais caros computacionalmente.

”

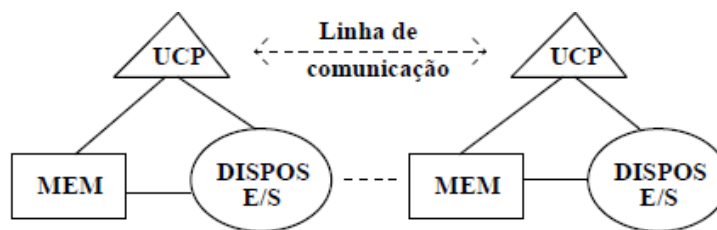
Multiprocessadores: sistemas fortemente acoplados

- Memória única.
- Tudo gerenciado por um único sistema operativo.
- Subdividido em
 - Arquitectura simétrica.
 - Assimétrico
- Custo de produção mais elevado.



Multi-processadores: sistemas fracamente acoplados

- Um único sistema operativo ou vários em vários nós
- Cada membro do sistema esta conectado aos outros por um link de dados
- Podemos ter sistemas sistemas de rede distribuídos
- Sistemas operativos de rede:
 - Requer o conhecimento prévio da localização de recursos
 - Cada nó poderá correr o seu sistema operativo
 - Em caso de falha de um dos nós, os serviços naquele nó ficam indisponíveis
- Sistemas operativos distribuídos:
 - Partilha de recursos e redundância de serviços
 - Controlo distribuído e transparência aos acesso de recursos
 - Mesmo sistema operativo em todos nós envolvidos



4. Concorrência e Multiprogramação

Computador - seu uso é caro computacionalmente

Objectivo: minimizar o desperdício (tempo que a UCP fica parada)

Concorrência: Conjunto de técnicas que permite a UCP passar mais tempo ocupada - desempenhando mais tarefas por unidade de tempo (*throughput*); representa a ilusão do paralelismo na execução das tarefas.

Momentos de desperdício:

- Frequência da UCP é maior do que a frequência da memória
- Velocidade da UCP é muito maior do que a velocidade de E/S.
- Baixo uso da UCP em si.

Solução: Manter a UCP o mais ocupada possível, executando instruções de forma concorrente (Multiprogramação)

A seguir algumas técnicas de concorrência:

Interrupções

Durante a execução, eventos inesperados podem ocorrer desvio forçado na execução - interrupção.

- Evento externo ao programa.
- Evento gerado por software (System call) ou hardware (IRQ- Interrupt Request)
- Fim do timeslice

A interrupção é o fundamento básico de concorrência, utilizado como linha de comunicação com o processador.

Arquitetura IBM-PC: 16 interrupções de hardware, divididas entre 2 controladoras de interrupção (interligadas em cascata):

Exemplo: IRQ1 - relógio (timer); IRQ3 - porta serial 1; IRQ7 - porta paralela; IRQ13 - IDE 1; etc.

Exceções

Parecidas com as interrupções, mas diferem nos requisitos:

- Evento interno ao programa.
- Tratamento é todo via software.

Exemplo: Divisão por zero, buffer overflow / underflow, stack overflow, etc.

Tratamento feito pelo desenvolvedor ou pelo hardware (em alguns casos):

- Falhas em programas geram exceções.
- Emergência de hardware como é caso de *reset*.

Controladoras de E/S e Técnicas de acesso aos dispositivos

- UCP comunica directamente com os dispositivos de E/S.
 - Problema: E/S é muito mais lento do que a memória ou a UCP.
- Introdução do controlador (ou controladora) de E/S:
 - Gestão do acesso aos dispositivos de E/S.
 - Tira da UCP o trabalho de cuidar dos dispositivos.
- Mais eficiente do que a UCP fazer o controle.

Problema ainda existe na transferência de dados entre a memória e os dispositivo de E/S que é feita pela UCP. Solução: Técnicas de acesso

- **E/S Controlada por Programa ou E/S Programada:** Implementação por software; CPU focada da operação de E/S, despoleta a operação e fica constantemente virada para mesmo, inquirindo o dispositivo de sobre a sua conclusão. Por isso mesmo empregue em ambientes monotarefa.
- **E/S Controlada por Interrupção ou E/S por Interrupção:** Suporta a multitarefa- a CPU despoleta a operação de E/S e enquanto o dispositivo de E/S não estiver *ready*, a CPU executa outras tarefa; No fim o dispositivo notifica a CPU- Interrupção.
- **E/S por DMA (Direct Memory Access):** A E/S por interrupção, revela-se ineficiente quando se trata de grandes transferências de dados pois a cada bloco de dados lidos deve-se notificar a CPU, o que acaba desperdiçando ciclos de processamento. O controlador DMA assume o papel da CPU no controlo das transferências e só no fim é que devolve o controlo ao processador. Assim a temos o processador à disposição para atender outras tarefas – suporte alargado à multitarefa.

Buffering

- Velocidade da UCP é muito maior do que a velocidade de E/S.
- Uso de uma área na memória principal como área de transferência intermédia: *buffer*.
- Permite acelerar o acesso aos dispositivos de E/S (leitura e gravação), matendo a CPU e E/S ocupados na maior parte do tempo.

Spooling

- Fila de submissão de tarefas - inicialmente em fitas magnéticas
- Ficheiro de *pool* - usado inicialmente em sistemas operativos do tipo batch
- Fila: 1º a entrar, 1º a sair (FIFO)
- Uso no gerenciamento de impressão hoje em dia

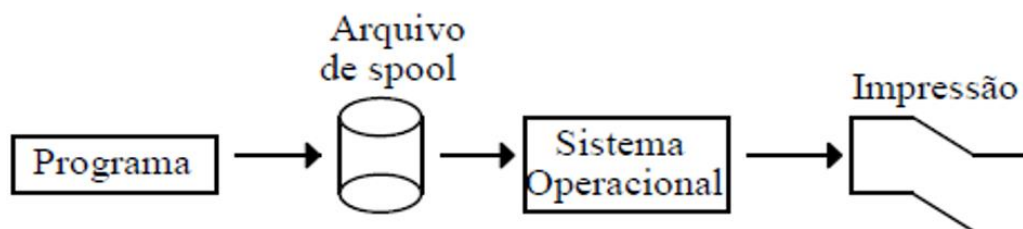


Fig.6 ∴ Implementação do Spooling

Reentrância

- Mais comum em sistemas multi-usuário.
- Vários usuários usando os mesmos programas.
- Problema: Várias cópias do mesmo programa ou módulos na memória provocam desperdício.
- Reentrância: capacidade do código do programa ser reutilizado e compartilhado entre os usuários do sistema
- Código re-executável = código reentrante.
- O código deve ter compilado com essa opção.
- Uso mais eficiente da memória e aumento do desempenho do sistema.

Segurança e Protecção do Sistema

- Sistemas recentes - mais complexos. Necessidade de aumentar a segurança
- É preciso garantir a confiabilidade e integridade de programas e dados
- Situações em que mecanismos de protecção são necessários :Evitar GPF; Comunicação entre programas de forma sincronizada; Evitar “monopólio” da UCP e contornar as excepções.
- Mecanismos básicos:
 - Modos de operação da CPU
 - Interrupção
 - Protecção, periféricos, memória e da CPU.

Modos de operação da CPU

- Os processadores modernos distinguem dois modos de operação: Modo kernel (Privilegiado) e Modo User (Não-privilegiado)
- No modo Kernel, o processador executa todas as instruções do processador, código do sistema operativo e drivers, isto é que tem acesso ao hardware
- Instruções a nível de aplicações, faz com que o processador passe a operar no modo User.

Interrupção

- Hardware: ocorrência de um evento externo
- Software: execução de uma instrução específica (system calls)
- Excepção: Erros de execução (overflow, underflow, etc)

Protecção de periféricos

- Instruções de E/S são privilegiadas

- Como os processos dos usuários realizam operações de E/S sendo estas Privilegiadas?
Solução: System calls ou Trap (método utilizado pelo processo de usuário para solicitar serviços ao Sistema Operativo.

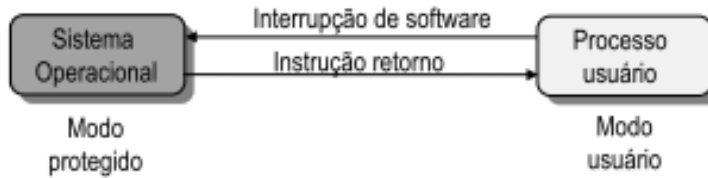


Fig.7 ∴ System call

Proteção de memória

- É necessário evitar-se situações de GPF (General protection Fault): isolamento dos espaços de endereçamento na memória.
- A faixa de endereçamento é delimitada conforme a figura que se segue

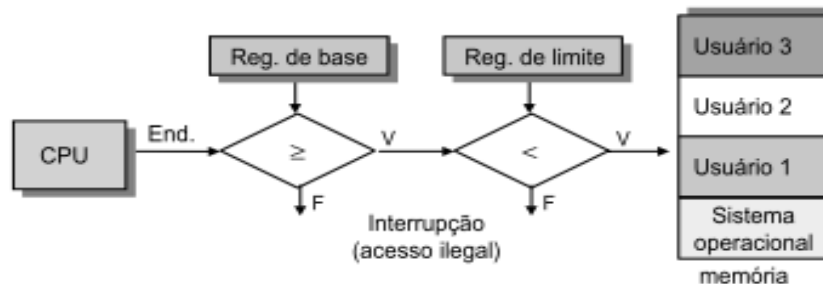


Fig.8 ∴ Proteção da memória

Proteção do processador

- E modo a garantir a execução do código do sistema operativo, deve correr de forma periódica uma interrupção de tempo (timer): para multiprogramação e contabilização do tempo
- Instruções relacionadas com a programação do tempo são privilegiadas

Chaveamento de modos (ou mudança de contexto) dos estados do processador, do modo Kernel para o *User*, e vice-versa. Custa (pouco) tempo à UCP, mas é desejável que esse tempo seja ainda mais minimizado.

Interrupção: Ocorre o chaveamento do modo User para o modo Kernel.

Execução de uma instrução: A CPU comuta do modo Kernel para modo User

5. Estrutura de um Sistema Operativo

Um Sistema operativo é Composto de três partes funcionais:

- Kernel;
- Bibliotecas; e
- Utilitários.

Kernel

- Núcleo do Sistema Operativo-parte central (núcleo)
- Oferece funções e serviços para se usar a CPU, memória primária e outros dispositivos

Como o kernel comunica com hardware e software, sua organização interna varia de projecto para projecto. E daí ser necessário que seja pequeno, rápido, estável e seguro.

Principais funções do kernel

- Gestão da CPU / processos (Programas em execução)
- Gestão de memória principal e secundária
- Gestão de dispositivos de E/S
- Gestão do sistema de ficheiros
- Tratamento de interrupções.
- Tratamento de excepções
- Suporte aos serviços de rede
- Implementação de mecanismos de segurança
- Contabilidade e auditoria do sistema (logging)

Bibliotecas

- Conjunto de rotinas usadas por programas.
- Fornecem serviços para os programas.
- Contém as chamadas ao sistema.

Chamadas do sistema (*system calls*)

- Partes integrantes das bibliotecas.
- Meio organizado e padronizado para acesso ao kernel.
- Forma como o kernel pode ser acessado.
- Esconde a complexidade do acesso para o programador.
- Programa / Aplicação > System Calls > Kernel > Hardware

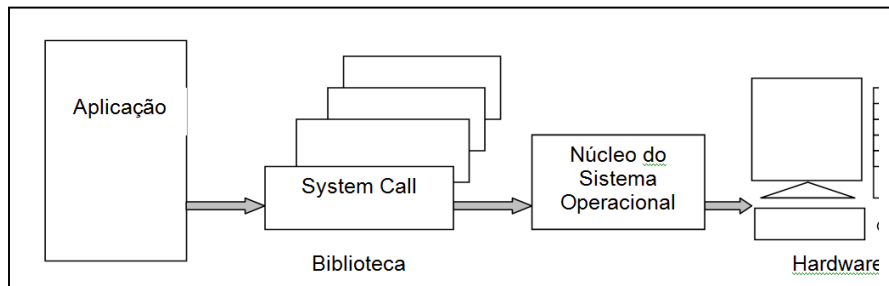


Fig.5 .:. *System call*

Utilitários

- Programas que auxiliam o funcionamento do sistema operativo.
- Exemplos: Compiladores, compactadores, ferramentas de acesso ao disco, etc.

6. Componentes e alternativas ao Sistema Operativo

Independentemente da sofisticação ou das capacidades do computador, um Sistema Operativo deve atender aos seguintes princípios:

- Oferecer os recursos do sistema de forma simples e transparente;
- Gerir a utilização dos recursos existentes buscando seu uso eficiente;
- Garantir a integridade e a segurança dos dados armazenados e processados no sistema e também de seus recursos físicos.
- Criar uma máquina virtual sobre a máquina física que ofereça todos recursos lógicos necessários para o desenvolvimento de aplicações (Fig.6)

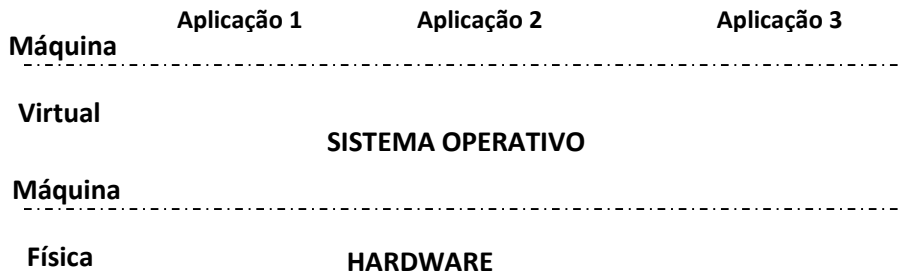


Fig.6 .:. Sistema Operativo como intermediário

Funções devidamente agrupadas, de acordo com a finalidade formam as componentes do Sistema Operativo, a saber:

Gestão de processos

Um processo abstração é um programa em execução. Um processo precisa de certos recursos, que inclui tempo de CPU, memória, ficheiros, dispositivos de E/S, para realizar suas tarefas.

- O SO é responsável pelas seguintes actividades com relação a gestão de processos:
 - Criação e eliminação de Processos.
 - Suspensão e resumo de processos.
- Fornecimento de mecanismos para:
 - Sincronização de processo
 - Comunicação inter processo (IPC- Interprocess Communication)

Gestão da memória principal

- Memória é um grande *array* de bytes, cada um com o seu endereço. É um depósito de acesso rápido de dados compartilhados entre CPU e E/S. É um dispositivo de armazenamento volátil.
- O SO é responsável pelas seguintes actividades com relação a gestão de memória:
 - Manter informações de que partes da memória estão em uso e por quem
 - Decidir que processos carregar quando espaços de memória estão disponíveis
 - Alocar e liberar espaço de memória quando necessário
 - Simular a memória em espaço do disco – memória virtual

Gestão do Sistema de ficheiros

- Um ficheiro é uma colecção de informações relacionadas definidas pelo seu criador. Ficheiros representam programas e dados.
- O SO é responsável pelas seguintes actividades com relação a gestão de ficheiros:
 - Criação e eliminação de ficheiro.
 - Criação e eliminação de directorias / pastas.
 - Suporte de primitivas para manipular ficheiros e directorias.
 - Mapeamento de arquivos na memória secundária.
 - Backup de ficheiros (cópia de segurança) em meios de armazenagem estáveis (não volátil).

Gestão do sistema de E/S

Um dos objectivos do SO é ocultar as peculiaridades de hardware específicos dos usuários. A gestão do sistema de E/S consiste em:

:

- Implementar áreas de transferência intermediária (buffer, caching e spooling)
- Criar uma interface geral para drivers de dispositivos
- Prover drivers genéricos para os dispositivos de hardware específicos

Um facto importante em relação ao sistema é que apenas o driver de dispositivo conhece as peculiaridades do dispositivo ao qual foi atribuído.

Gestão de armazenamento secundário

Memória principal é volátil e pequena, sistema deve fornecer memória secundária para funcionar como backup da principal. A maioria dos sistemas modernos usa discos. O SO é responsável pelas seguintes actividades com relação a gestão de disco:

- Gestão de espaço livre
- Alocação de espaço
- Escalonamento de disco (gerir as filas das requisições, incluindo a optimização)

Segurança e Protecção de sistema

- Protecção refere-se a um mecanismo para controle de acesso de programas e outros recursos
- O mecanismo de protecção deve:
 - Distinguir entre uso autorizado e não autorizado.
 - Especificar os controles a serem impostos.
 - Fornecer um meio de forçar os mesmos.

Gestão de rede

- Implementa serviços e protocolos de rede, oferecendo mecanismos configuração e optimização dos sistemas.

Interpretador de comandos

- Um dos programas de sistema mais importantes para um sistema operativo é o interpretador de comandos, que é a interface entre o usuário e Sistema Operativo.
- Alguns Sistemas Operativos incluem o Interpretador de comandos no kernel. Outros, tais como MS-DOS e o UNIX, tratam o ele como um programa especial que fica executando quando uma tarefa é iniciado ou quando um usuário entra no sistema.
- Por exemplo, os comandos para as seguintes actividades podem ser dados ao SO no interpretador de comandos:
 - Criação de processos e gestão
 - Tratamento de E/S
 - Gestão de memória secundária
 - Gestão de memória principal
 - Acesso ao sistema de ficheiros
 - Protecção
 - Rede
- O programa que lê e interpreta comandos de controle é chamado interpretador de linha de comandos. No UNIX/LINUX é o shell. Sua função é capturar e executar o próximo comando de controle.

Alternativas ao Sistema Operativo

As linguagens de programação podem produzir todo código necessário para que um programa se execute directamente sobre o hardware. Entretanto, isso implicaria:

- O esforço de programação seria muito grande;
- Um conjunto significativo de funções seria repetido;
- Cada aplicativo poderia optimizar o seu desempenho, mas globalmente a máquina ficaria subaproveitada;
- Não seria possível ter políticas globais de segurança, tolerância à falhas e optimização.

7. Atributos de qualidade de um Sistema Operativo

Eis alguns dos atributos que podemos ter em conta na escolha de um sistema operativo:

Desempenho

- Gestão eficiente dos recursos físicos que suportam os lógicos.

Segurança

- Isolamento dos utilizadores
- Permitir a partilha segura dos recursos lógicos.

Fiabilidade

- Detectar um conjunto de falhas.
- Tolerar um conjunto de erros.

Interface de Programação Completa e Simples

- Facilitar a concepção de aplicações, sua manutenção e portabilidade.

Interface de operação e gestão de recursos lógicos de fácil utilização

8. Arquitectura e Classificação de Sistemas Operativos

Do ponto de vista da sua arquitectura os sistemas operativos podem ser vistos como Monolíticos, Dispostos em camadas, Máquina virtual e Mikrokernel (Cliente-Servidor).

Monolítico

- Um grande bloco de código, ou dividido em módulos, todos sendo executados no modo kernel.
- Estrutura mais simples.
- A manutenção pode ser complexa, se não for um projecto bem feito.
- Simples na estrutura.
- Rápido devido a sua estrutura simples
Ex: MS-DOS e algumas distribuições Linux.

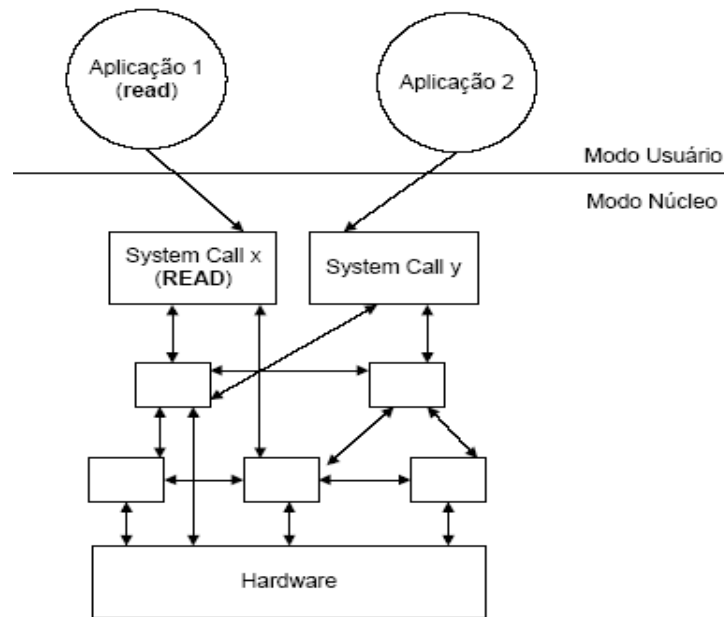


Fig.8 .. Sistema Operativo monolítico

Disposto em camadas ou níveis

- Níveis sobrepostos.
- Há necessidade de passar por todos os níveis para chegar ao kernel.
- Como as camadas são isoladas, facilita a manutenção e a depuração.
- Desempenho prejudicado pela estrutura - muito burocrático.

Ex: Open VMS, MULTICS, Windows 2000

5	Operador
4	Programas de Usuário
3	Entrada/Saída (E/S)
2	Comunicação
1	Gerenciamento de Memória
0	Multiprogramação

Fig.9 .. Sistema Operativo em níveis

Máquina Virtual

- Sistema computacional composto por níveis, onde o nível mais baixo é o hardware.
- Modelo de máquina virtual - nível intermediário entre o hardware e o sistema operativo – gestão de Máquinas virtuais.
- Cada máquina virtual oferece uma cópia virtual do hardware, incluindo os modos de acesso, interrupções, dispositivos de E/S, etc.
- Cada máquina virtual é independente das demais, contendo seu próprio Sistema Operativo, seus próprios usuários e suas próprias aplicações.
- Conceito iniciado no VM/370, baseado no OS/370, da IBM (anos 1960).
- Conceito usado também com a linguagem Java - JVM.

Vantagens:

- Segurança - Isolamento total das máquinas virtuais.
- Economia de recursos – é mais barato um servidor grande do que vários servidores pequenos.
- Portabilidade - se o hardware hospedeiro tiver um defeito, basta transferir os ficheiros das máquinas virtuais para outro hardware.

Desvantagens:

- Complexidade.
- Sobrecarga - um hipervisor é complexo e consome muitos recursos do hardware hospedeiro.

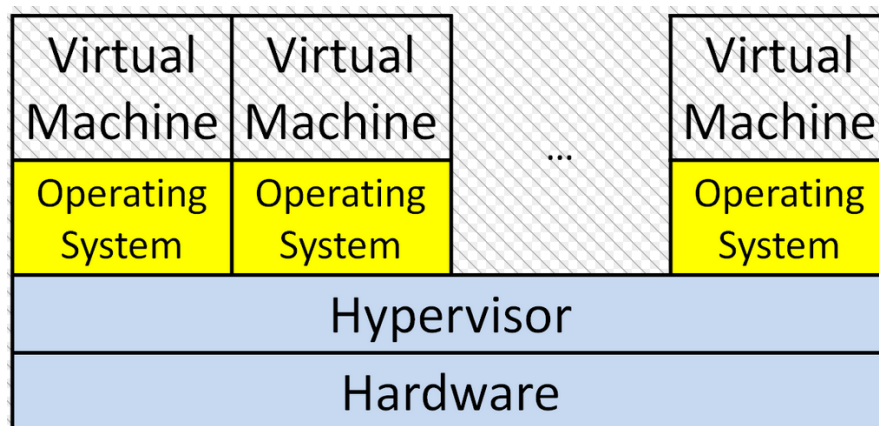


Fig.10 .:. Modelo de Máquina Virtual Tipo I

Microkernel

- Constatação - sistemas actuais ainda são lentos e pesados.
- Microkernel - tornar o núcleo do sistema operativo o menor e mais simples possível.
- Disponibilizar os serviços através de processos a serem executados no nível usuário.
- Cada um desses processos servidores fornece um recurso específico para o sistema: gestão de processos, gestão de ficheiros, escalonamento de processos, etc.
- A principal função do kernel então é fazer o diálogo entre os diferentes processos servidores.

Vantagens:

- Mais seguro: Se um serviço sair do ar, é recolocado facilmente.
- Adequado para computação distribuída: Os serviços podem ser remanejados entre as UCPs.
- Mais flexível.
- Mais rápido: com o kernel enxuto, menos código a ser executado.
- Mais facilmente condensado para outras arquiteturas: Menos código no kernel, menos dificuldades para reescrever o kernel para outras plataformas.

Desvantagens:

- As mudanças constantes entre os modos usuários e kernel podem comprometer a performance do sistema, uma vez que a dado momento processos de usuários irão solicitar em demasia a atenção do kernel, forçando a CPU a fazer mudanças constantes no modo de funcionamento, ora modo kernel ora usuário.

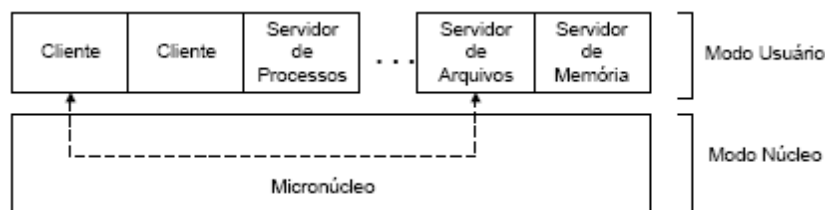
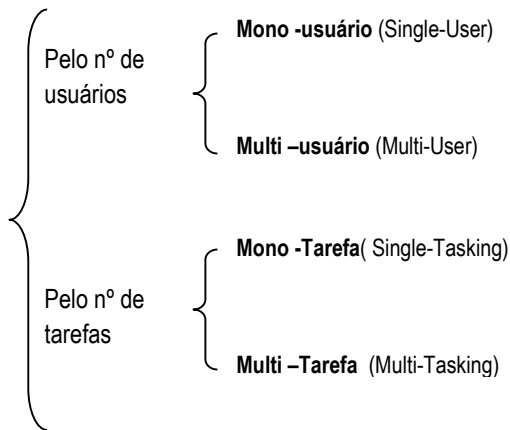


Fig.11 ∴ Modelo Cliente-Servidor

Classificação de Sistemas Operativos

Vamos aqui considerar outras formas de classificação de Sitemas Operativos: (1) Pelo Serviço e (2) Interface de utilizador e (3) Mistos.

1. Pelo serviço



2. Interface de utilizador

Linha de comandos

Alguns Sistemas operativos apenas fornecem interface de usuário baseado em texto, isto é, linha de comandos (CLI - *Command line interface*)

Esta forma de trabalhar exige um conhecimento prévio dos comandos e da respectiva sintaxe

Exemplo: **C:\> COPY A:\AULA.TXT C:**

Interface gráfica (GUI – Graphical User Interface)

Outros sistemas operativos são concebidos de modo fornecer um ambiente gráfico e amigável (*User-friendly*), baseado em objectos (ícones, janela, caixas de diálogos, etc.) de fácil utilização e que seja intuitivo.

3. Mistos

Mono - Usuário, Mono -Tarefa

- Apenas um utilizador de cada vez a executar uma única tarefa (aplicação)

- Exemplo: Palm OS (em computadores Palm Top)
Exemplo: Java OS

Mono - Usuário, Multi -Tarefa

- Apenas um usuário com vários programas a correm em simultâneo
- Exemplos: Ms Windows 98, primeiras versões de MAC OS.

Sistema Operativo. Definição

Algumas definições importantes de autores da área:

[1] *Sistema Operativo são programas, implementados tanto em software quanto em firmware, que tornam o hardware usável. O SO torna o poder computacional do hardware disponível convenientemente para o usuário e gere o hardware cuidadosamente a fim de obter boa performance.*

[2] *Um Sistema Operativo é um programa que age como um intermediário entre o usuário do computador e o hardware. O propósito de um SO, é prover um ambiente no qual um usuário possa executar programas de forma conveniente e eficiente.*

[3] *Um Sistema Operativo é um conjunto de módulos de software que colectivamente servem como interface entre programas aplicativos e os sistemas de hardware, com o objectivo fornecer um sistema computacional eficiente, seguro e fácil de usar.*

[4] *Um Sistema Operativo pode ser definido como a parte do computador que transforma o hardware sem vida em um sistema poderoso e usável.*