SISTEMAS OPERATIVOS

Evolução dos sistemas operativos

António Godinho

O QUE É UM SISTEMA OPERATIVO?

Perspetiva de máquina virtual:

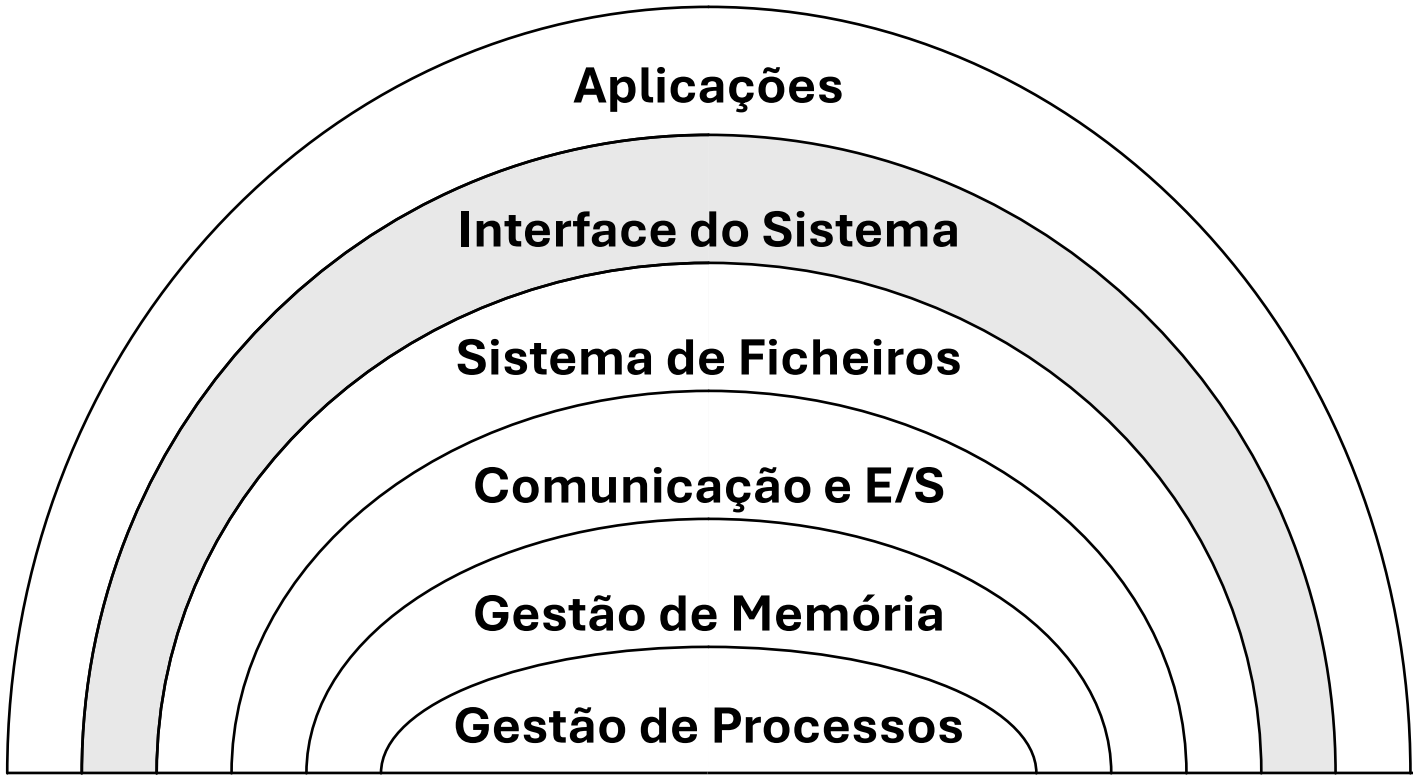
- SO é uma extensão do hardware que implementa uma interface para as aplicações.

Perspetiva de gestor de recursos:

- SO é um gestor dos recursos físicos e lógicos do sistema.

  - Recursos físicos (hardware): processador, memória, dispositivos de entrada/saída(E/S), discos, terminais, etc.
  - Recursos lógicos (software): programas, ficheiros, base de dados, interfaces com o utilizador, etc. Os recursos lógicos são abstrações definidas de forma a aproximar as entidades do mundo real, que se pretendem automatizar, dos sistemas computacionais.

ESTRUTURA EM CAMADAS



As principais funções do sistema encontram-se associadas a uma camada

PERSPECTIVAS SOBRE UM SO

Ponto de vista do utilizador

- Interface de utilização: interface por linha de comandos (shell) ou interface gráfica (GUI-Graphical User Interface).

- Sistemas de ficheiros: gestão hierarquizada de diretorias, ficheiros de aplicações e dados.

- Segurança e proteção de aplicações e dados: cada utilizador é identificado no SO, obtendo uma conta de acesso à qual está associado um perfil de utilização. Este perfil, em conjugação com as permissões associadas a cada recurso (ex. diretorias e ficheiros), permite implementar um mecanismo de segurança e proteção de aplicações e dados.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do utilizador

▪Programas: Em execução, são suportados por uma entidade lógica designada por processo. É possível a execução de múltiplos processos do mesmo programa. Programas aplicacionais interativos; Programas de sistema (normalmente não interativos e executados em segundo plano - background);Programas de administração; Utilitários do SO.

▪Memória virtual: permite a execução de múltiplos programas/processos. Quando a memória física não é suficiente, o SO mantém em memória apenas as secções de código e dados estritamente necessárias. As restantes secções são transferidas para disco (swapping) e recuperadas sempre que necessário. O SO usa o disco como expansão da memória física.

▪Periféricos: teclado, monitor, rato, impressora, etc.

▪Interface de rede: permite a comunicação de dados com outros computadores ligados à rede.

PERSPECTIVAS SOBRE UM SO

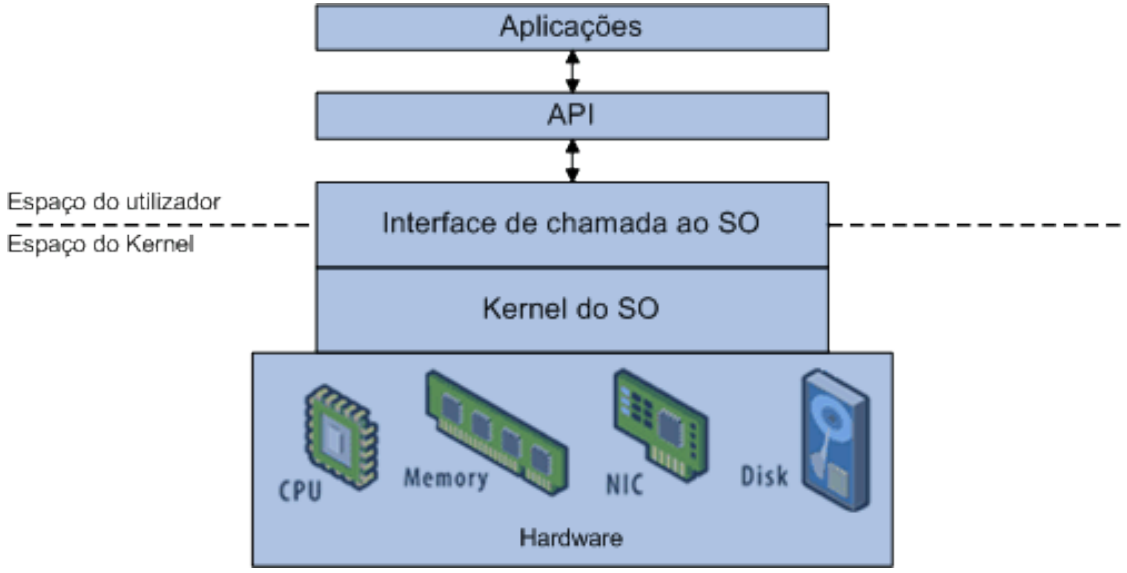
Ponto de vista do programador de aplicações

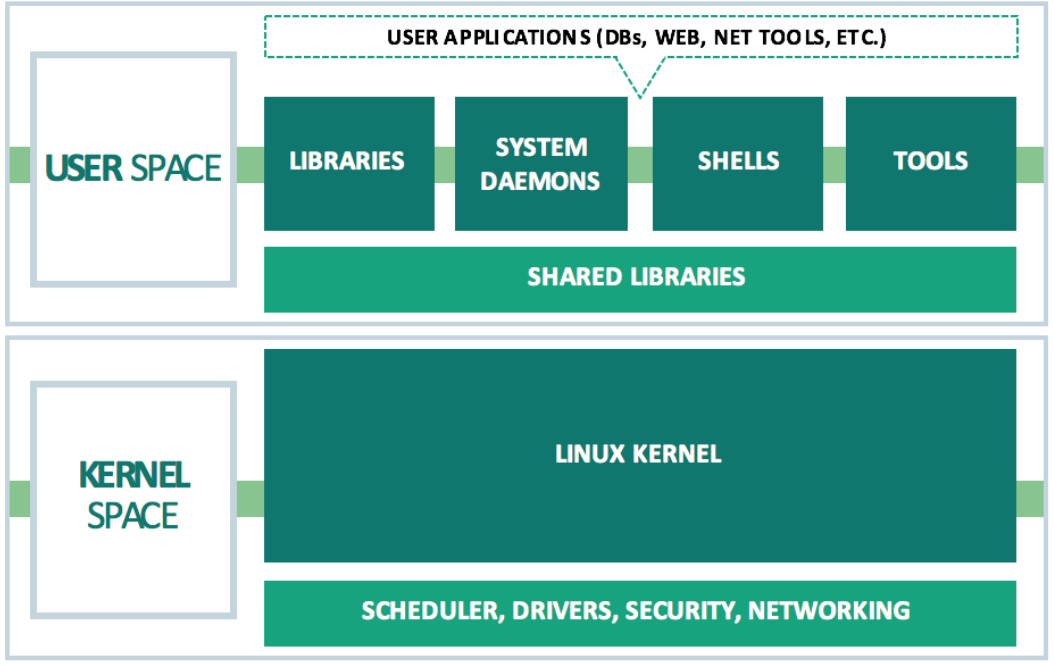
- Ferramentas de desenvolvimento: editor, compilador, linker, debugger.
- Bibliotecas de funções(Ex. libc standard library, DLLs).
- Interface coerente de programação (API - Application pogramming interface) para as aplicações, permitindo que estas acedam aos recursos lógicos do computador.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador de aplicações

- API: conjunto de rotinas que os programadores podem utilizar para requisitar serviços do SO. Os processos executam chamadas das funções definidas na API para aceder a serviços disponibilizados por camadas inferiores do sistema (chamadas ao sistema - system calls) (Ex. POSIX - Portable Operating System Interface e Win32 API). As aplicações assim desenvolvidas podem ser executadas em qualquer plataforma de hardware desde que tenha o mesmo SO (portabilidade).





PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador de aplicações

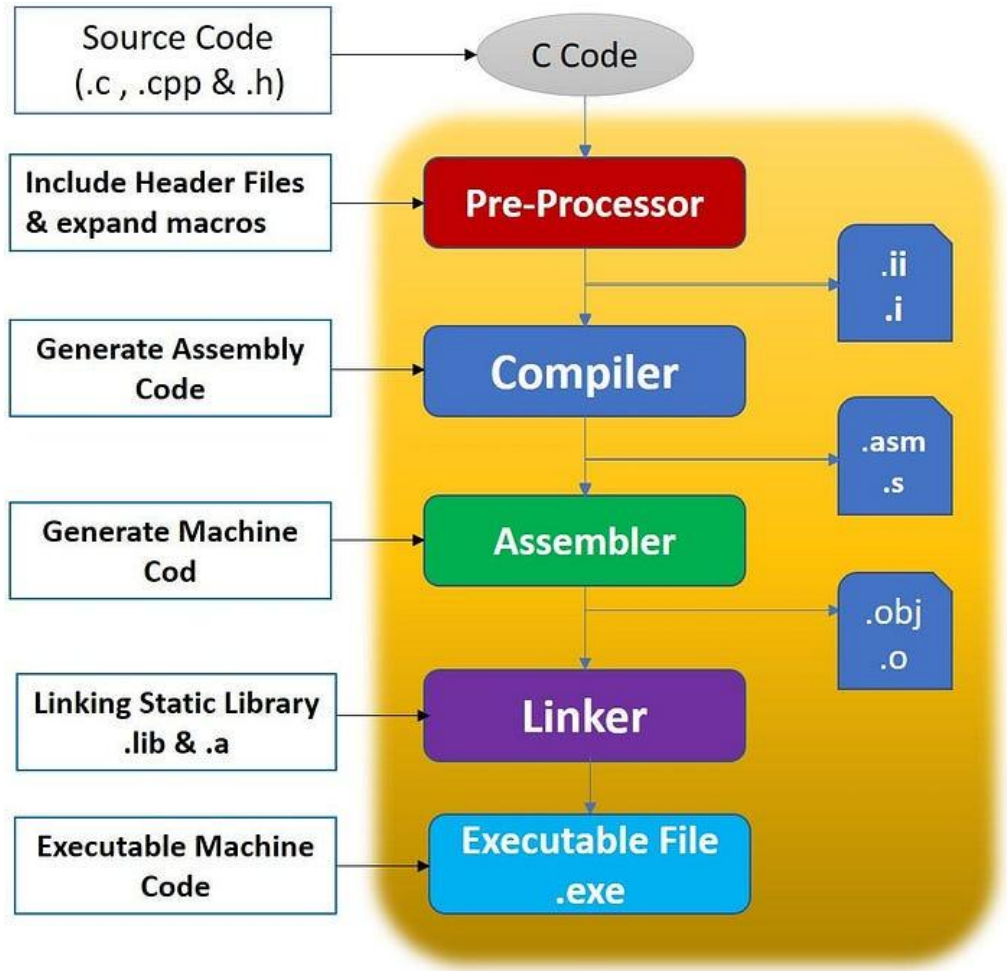
- API: conjunto de rotinas

Passo 1: Preprocessing

Passo 2: Compilation

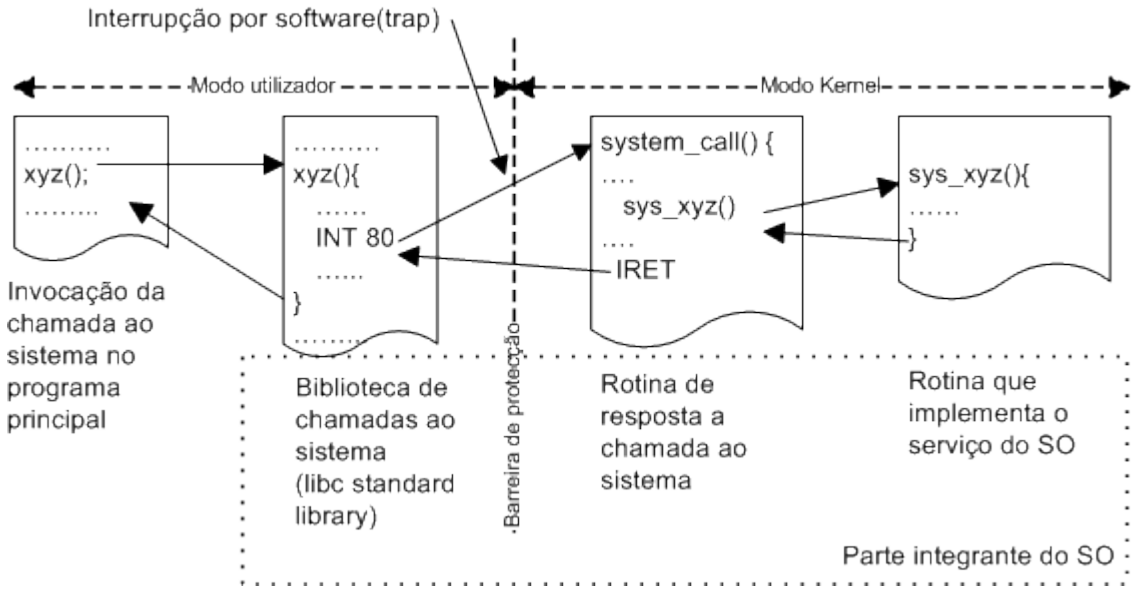
Passo 3: Assembly

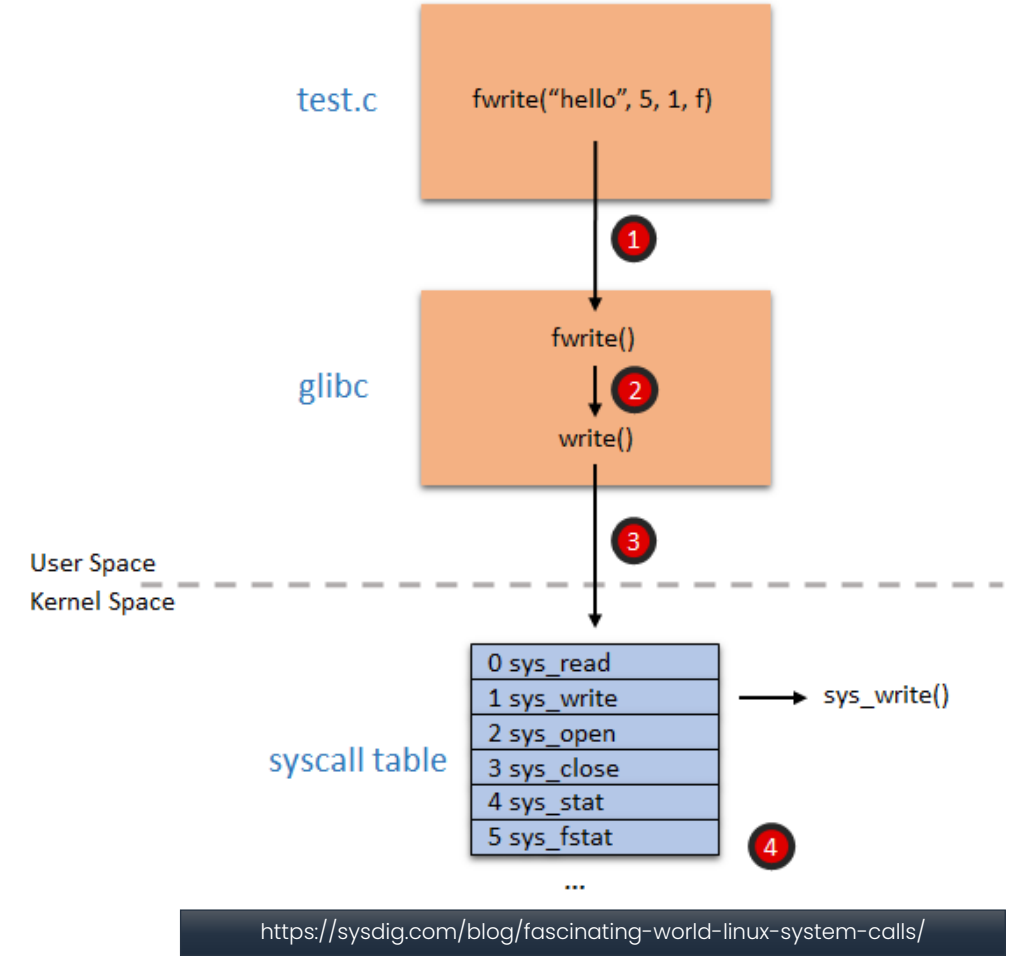
Passo 4: Linking



PERSPECTIVAS SOBRE UM SO

- Chamadas ao sistema (interface do SO):





A biblioteca de chamadas ao sistema é linkada com o código da aplicação. Cada função de sistema é identificada por um número que permite ao SO identificar o código que implementa o respectivo serviço. O código da biblioteca define o número da função de sistema e invoca a função system call(), através de uma interrupção por software (trap). Neste instante, a CPU muda do modo de execução de utilizador para kernel, permitindo o acesso às estruturas de dados do SO de modo seguro (pelo código do próprio SO).

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador de aplicações

- Chamadas ao sistema (interface do SO):
  - O código das funções de sistema são partilhadas por todos os processos.
  - A modificação das funções de sistema é transparente para todas as aplicações desde que a interface se mantenha.

```
// Exemplo de uma interrupção em C
#include <dos.h>
int main () {
        _AH = 0x0A;
        /* Especifica que queremos o serviço n.º 0Ah */
        _AL = '*';      /* (ASCII 42 dec = '*') */
        _BH = 0; /* Página de ecrã 0 */
        _CX = 10; /* Repete o caracter 10 vezes */
        geninterrupt (0x10); /* Chama a interrupção 10h. */
        return 0;
}
```

PERSPECTIVAS SOBRE UM SO

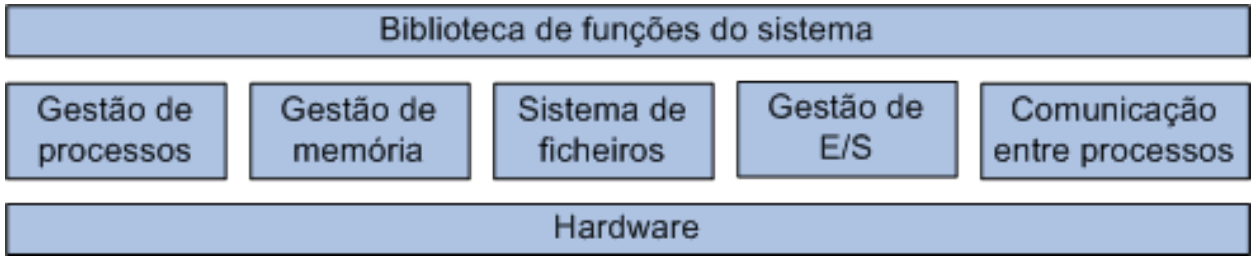
Ponto de vista do programador do SO: Objectivos na concepção do SO

- **Eficiência**: alto desempenho e tempos de resposta diminutos
- **Robustez**: tolerante a falhas e fiável - o SO não falha na totalidade devido a erros isolados de aplicações ou hardware. No caso de falha deve minimizar a perda de trabalho.
- **Escalabilidade**: capacidade de utilizar recursos à medida que estes são adicionados (ex - a adição de novos processadores deve traduzir-se num aumento proporcional de capacidade de processamento, proporcionalidade que nem sempre é conseguida).
- **Extensibilidade**: adaptável a novas tecnologias, permitindo a extensão a novas tarefas para além daquelas que inicialmente foram concebidas no SO.
- **Portabilidade**: concepção do SO te tal modo que possa operar em diferentes configurações de hardware.
- **Segurança e protecção**: não permite a utilizadores e software o acesso a serviços e recursos para os quais não tenham autorização. A protecção refere-se ao mecanismo que implementa a política de segurança do SO.
- **Interactividade**: permite que as aplicações respondam de forma rápida a eventos ou acções do utilizador.
- **Usabilidade**: potencial para servir uma base significativa de utilizadores e aplicações, recorrendo a interfaces interactivas e amigáveis.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Principais componentes

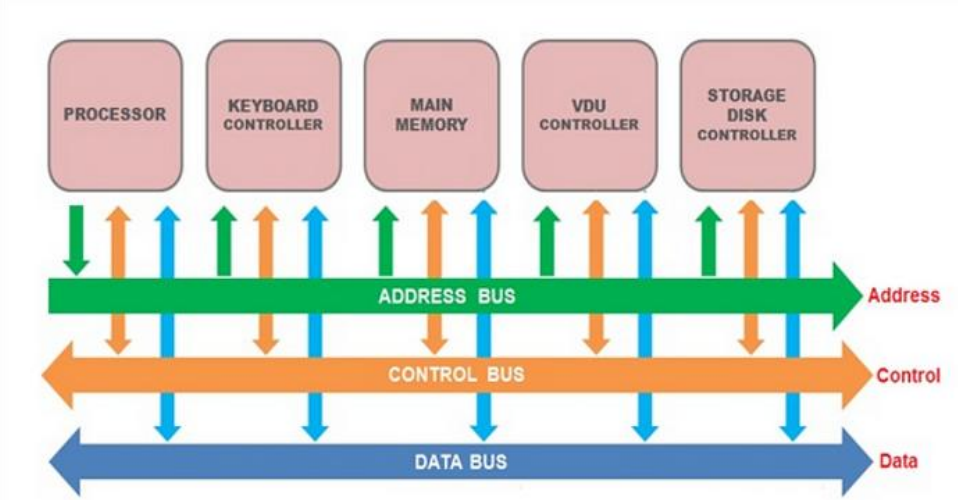
- Gestor de processador: implementa o mecanismo de escalonamento e despacho de processos, determinando quando e por quanto tempo um processo executa na CPU.
- Gestor de memória: determina quando e quanta memória é alocada a um processo.
- Gestor de E/S (periféricos): serve os pedidos de entrada e saída de e para os dispositivos periféricos de hardware, respectivamente.
- Gestor de comunicação entre processos: permite a comunicação e sincronização de actividades entre processos.
- Gestor do sistema de ficheiros: organiza uma colecção de dados nos dispositivos de memória secundária e proporciona uma interface coerente e consistente para aceder a esses dados.
- Bibliotecas de funções do sistema: implementa um conjunto de funções que permitem o acesso aos serviços do SO.

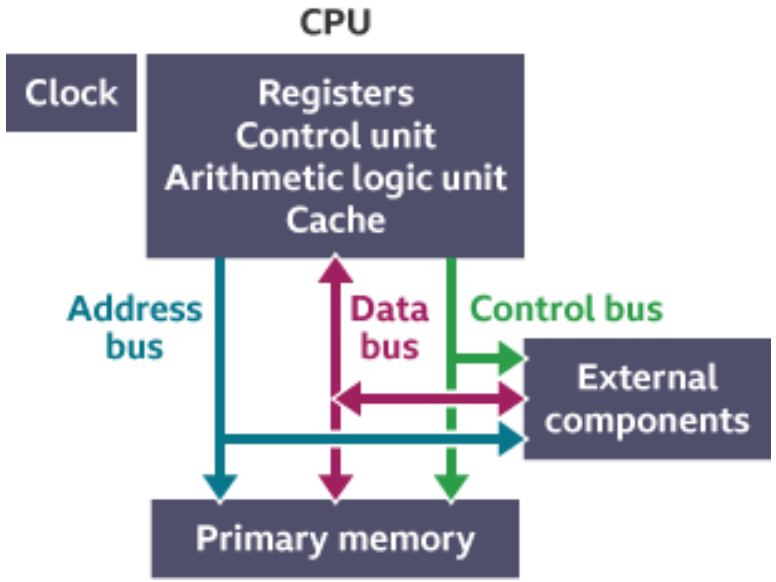


PERSPECTIVAS SOBRE UM SO

▪Ponto de vista do programador do SO: Principais elementos de hardware

- Estrutura geral do hardware
- CPU (processador)
  - Onde são executados os processos e o próprio SO
  - Constituído por uma unidade de busca/descodificação de instruções, unidade aritmética e lógica, registos e caches de dados e instruções
  - Registos: memórias que armazenam dados a serem processados e endereços para referenciação da memória (Ex. Program Counter PC- aponta o endereço de memória que armazena a próxima instrução a ser executada)
  - Possui pelo menos dois modos de funcionamento: user mode e kernel mode.





PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Principais elementos de hardware

- CPU (processador)

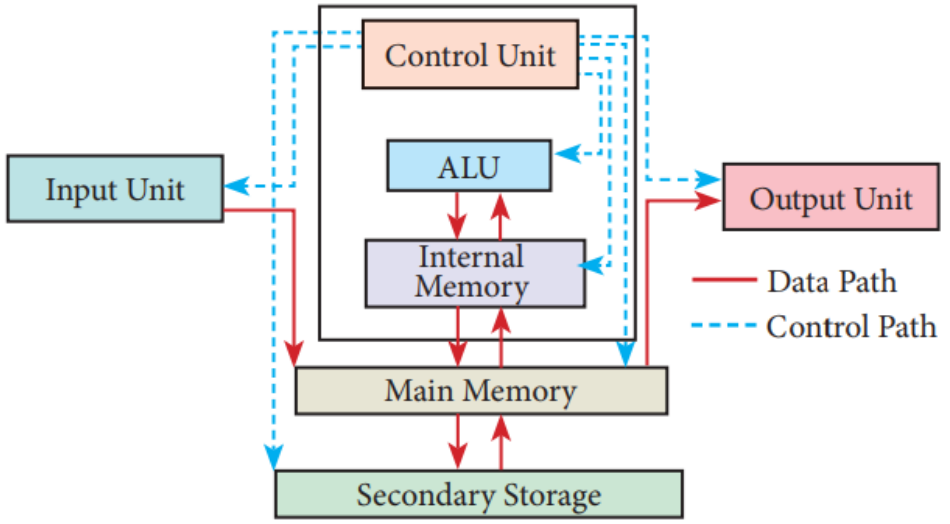
  - O SO garante que cada processo utiliza o processador durante o tempo que é necessário para o seu completamento.

  - A responsabilidade de escalonar os processos ao processador é do SO.

  - O processador executa apenas um processo de cada vez mas dá a ilusão que todos eles se executam em simultâneo.

  - Se um processo solicita uma operação de E/S, o SO atribui o processador a outro processo.

  - O processador é apenas interrompido quando ocorre uma interrupção ou exceção



PERSPECTIVAS SOBRE UM SO

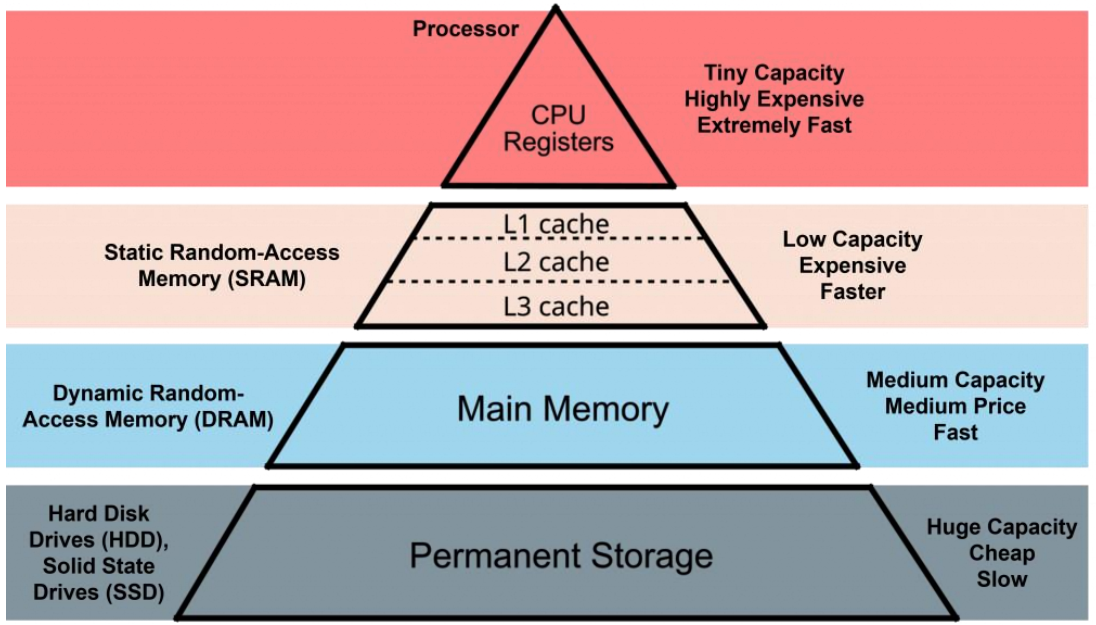
Ponto de vista do programador do SO: Gestão de memória

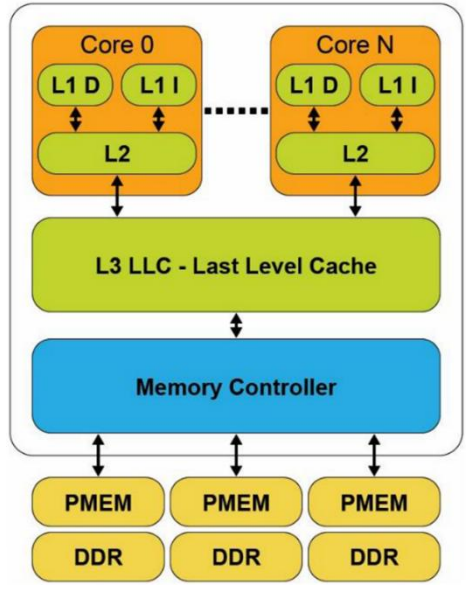
O SO garante que:

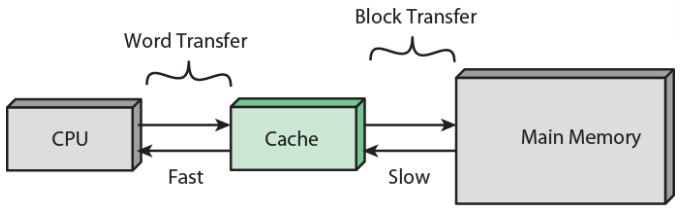
Os diferentes tipos de memória (RAM, cache, disco) são usados de forma a que cada processo se execute o mais rapidamente possível; Cada processo tem memória suficiente para ser executado; Os processos estão protegidos uns dos outros (isolamento: os processos não acedem à memória onde se encontram os dados de outros).

Memória virtual:

Os processos endereçam a memória de forma lógica (endereços virtuais). Os endereços virtuais são traduzidos em tempo de execução para endereços físicos. Nem todos os dados e código associados a um processo precisam de estar em memória principal (RAM) durante a sua execução. O código e os dados que não são necessários são guardados em disco (swapping). O SO garante que um dado processo apenas acede ao código e aos dados que estão no seu espaço de endereçamento







PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Interrupções e exceções

- Interrupções: indicam que um determinado evento ocorreu.

- Exceções: indicam que ocorreu um erro.

- Interrupções de E/S: A CPU é notificada pelos controladores de E/S de forma assíncrona. O processador:

  - termina a execução do processo, salvaguardando o seu contexto de execução em memória;

  - identifica o dispositivo que deu origem à interrupção e invoca, baseado no número da interrupção e na tabela de vetores de interrupção, a rotina de resposta a interrupção a ser executada

  - no fim da execução da rotina de resposta a interrupção, é reposto o contexto de execução do processo (ou de outro) e a execução continua.

- Interrupções do temporizador (Timer): São interrupções geradas periodicamente e utilizadas pelo SO para o escalonamento de processos. A cada processo é atribuído um intervalo de tempo (quantum ou time-slice).

- Interrupções por software (trap): invocadas pelo próprio processo através de instruções especiais do processador. Normalmente utilizadas nas chamadas ao sistema.

- Exceções: divisão por zero, execução de uma instrução ilegal, referência de uma posição de memória fora do espaço de endereçamento do processo, execução de uma instrução privilegiada quando em modo utilizador.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Gestão de E/S

- A interacção entre o SO e a unidade de E/S é feita através de um pequeno programa (device driver) que executa em modo kernel (Ex.um driver de disco aceita os dados de um ficheiro que o SO lhe envia e escreve o respectivo bloco de bytes no disco).

- O driver interage com o SO de modo a não ser necessário recompilar o SO quando um novo driver é instalado. A comunicação entre o SO e o driver é, em larga medida, baseado no mecanismo de interrupções.

- A maioria das unidades de E/S têm um controlador com buffers para fazer a adaptação, em termos de velocidade de funcionamento, entre o periférico e a CPU.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Gestão de E/S

- E/S programado (Polled I/O): O driver envia um comando ao controlador e, em ciclo, consulta o estado da unidade E/S para verificar se a operação está concluída. A CPU é responsável pela transferência de dados entre a RAM e o buffer do controlador de E/S.

- E/S por interrupções (Interrupt-driven I/O): O driver envia um comando ao controlador e liberta a CPU para executar outro processo. Quando termina a operação, o controlador notifica a CPU através de uma interrupção. O driver procede à transferência de dados entre RAM e E/S. A CPU é responsável pela transferência de dados entre a RAM e o buffer do controlador de E/S.

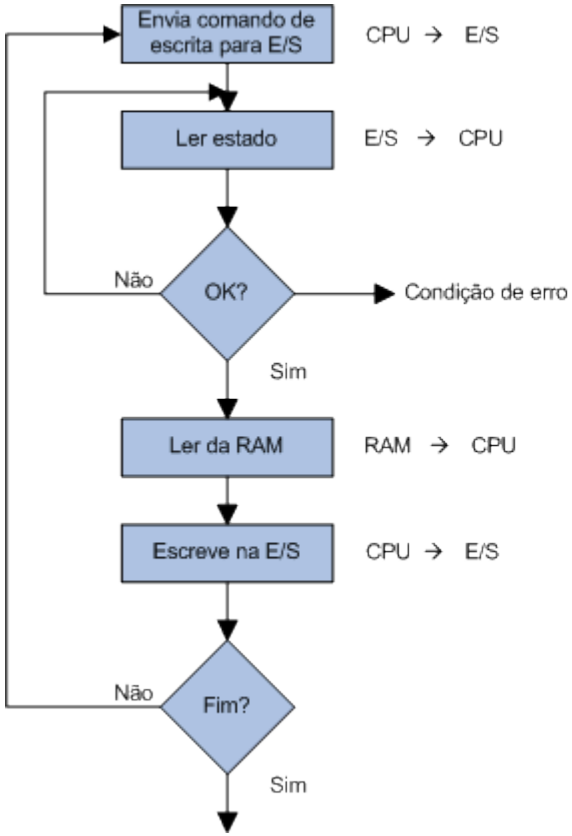
- E/S por DMA (Direct Memory Access I/O): O próprio controlador de E/S é responsável pela transferência de dados da RAM de e para o buffer do controlador de E/S. A CPU apenas intervém no início (envio de um comando de E/S para o controlador) e no fim da transferência (interrupção E/S enviada pelo controlador à CPU a indicar o fim da operação de E/S).

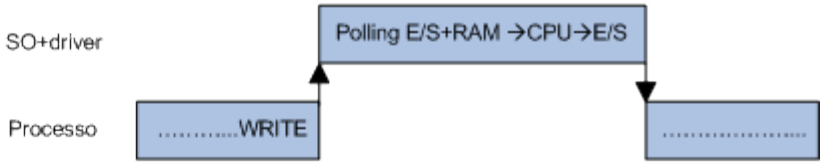
- Sistema de ficheiros: gestão hierarquizada de directorias e ficheiros de dados e aplicações. Interface uniforme de acesso a recursos (open, read, write, etc.). Implementado em dispositivos de memória secundária, onde os dados, aplicações e o próprio SO é guardado de forma persistente (discos, CD, DVD).

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: E/S programado

o módulo de E/S executa a ação solicitada e, em seguida, define os bits apropriados no registo de estado de E/S, mas não toma qualquer outra ação para alertar o processador (sem interrupção)





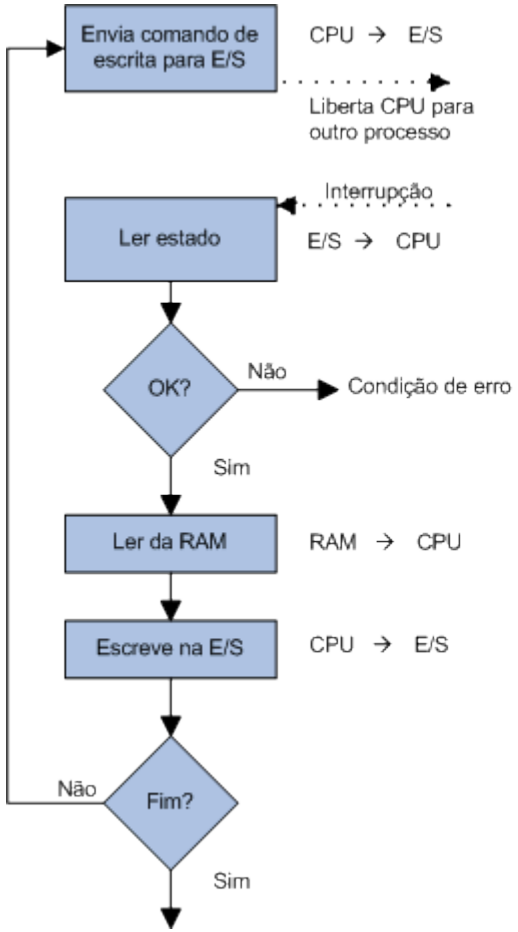
Com as E/S programado, o processador tem de esperar muito tempo até que o módulo de E/S em causa esteja pronto para receber ou transmitir mais dados. O processador, enquanto espera, tem de interrogar repetidamente o estado do módulo de E/S.

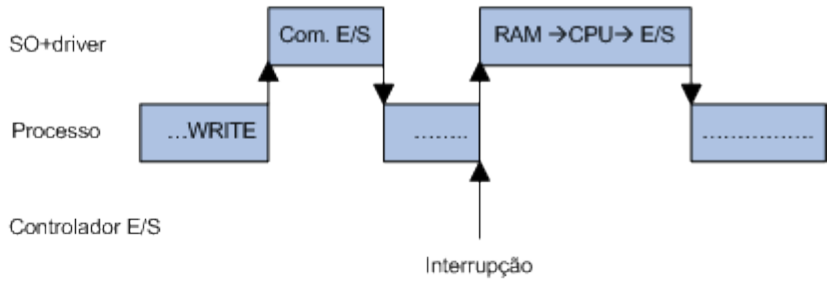
Como resultado, o nível de desempenho de todo o sistema é severamente degradado.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: E/S por interrupção

o módulo de E/S interrompe o processador para solicitar serviço quando estiver pronto para trocar dados com o processador. O processador executa então a transferência de dados, e retoma o seu processamento anterior.





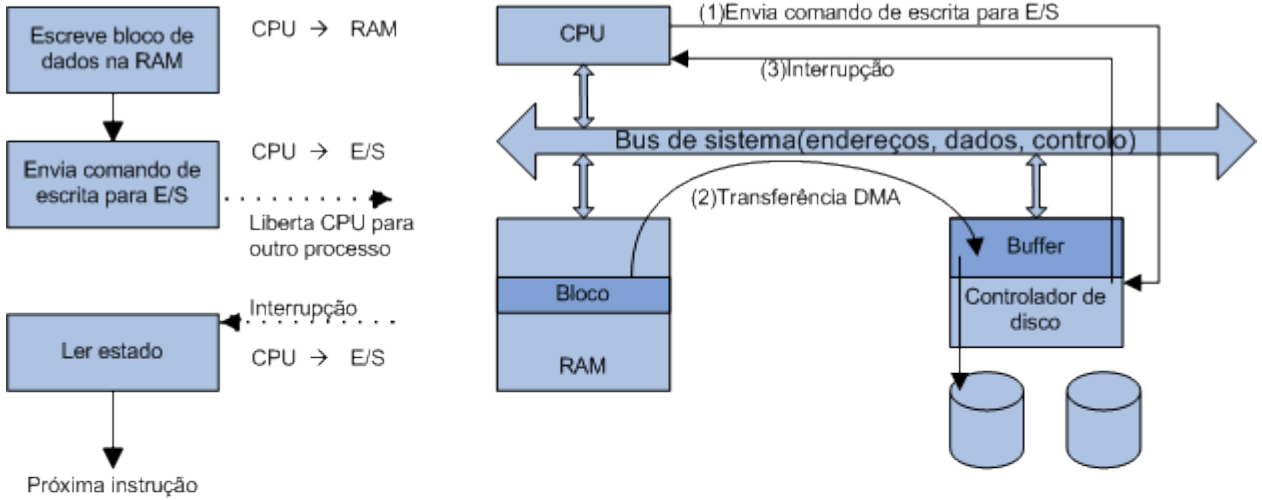
O mecanismo de interrupção elimina a necessidade de o processador verificar periodicamente as E/S (ou seja, elimina a necessidade de estar ocupado em espera).

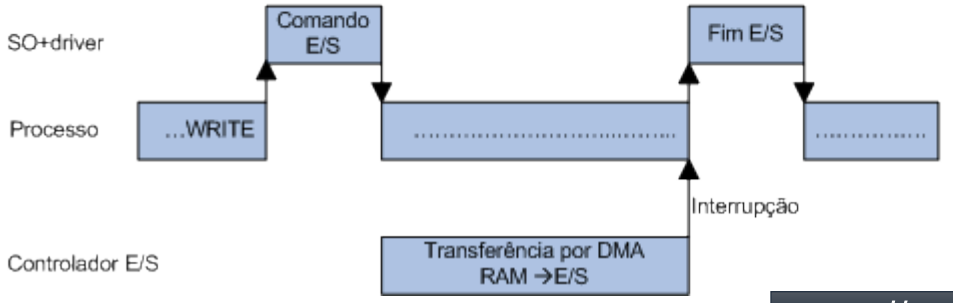
Isto liberta o processador para fazer outro trabalho, não precisa de verificar constantemente o módulo de E/SO processo será BLOQUEADO, o processador pode executar outros processos.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: E/S por Direct Memory Access (DMA)

O DMA permite que um dispositivo de E/S envie ou receba dados diretamente da ou para a memória principal contornando a CPU para acelerar as operações de memória.





https://stevengong.co/notes/Interrupt

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: E/S por Direct Memory Access (DMA)

O processador continua então a executar outra tarefa.

Delegou a operação de E/S ao módulo DMA - o módulo DMA transfere todo o bloco de dados, uma palavra de cada vez, diretamente para ou da memória, sem passar pelo processador. Quando a transferência é concluída, o módulo DMA envia um sinal de interrupção para o processador.

O processador só está envolvido no início e no fim da transferência.
Resumidamente, faz com que o processador execute mais lentamente durante uma transferência DMA, quando é necessário o acesso do processador ao barramento (controlado temporariamente pelo múdulo DMA). No entanto, para uma transferência de E/S de multiplas palavras, o DMA é muito mais eficiente do que a E/S programado ou acionado por interrupções.

PERSPECTIVAS SOBRE UM SO

Ponto de vista do programador do SO: Comunicação entre processos

- **Considerações na implementação de um canal de comunicação entre um processo produtor e um processo consumidor:** modo de transferência da mensagem e sincronização da comunicação.

- **Transferência da mensagem :** memória partilhada ou através do espaço de endereçamento do kernel do SO.

- **Memória partilhada:** os processos acedem a uma zona de memória que faz parte do espaço de endereçamento dos processos comunicantes

- **Espaço de endereçamento do kernel do SO:** os dados são sempre copiados para o núcleo antes de serem transferidos