Sistemas Operativos

Licenciatura Engenharia Informática
Licenciatura Engenharia Computacional

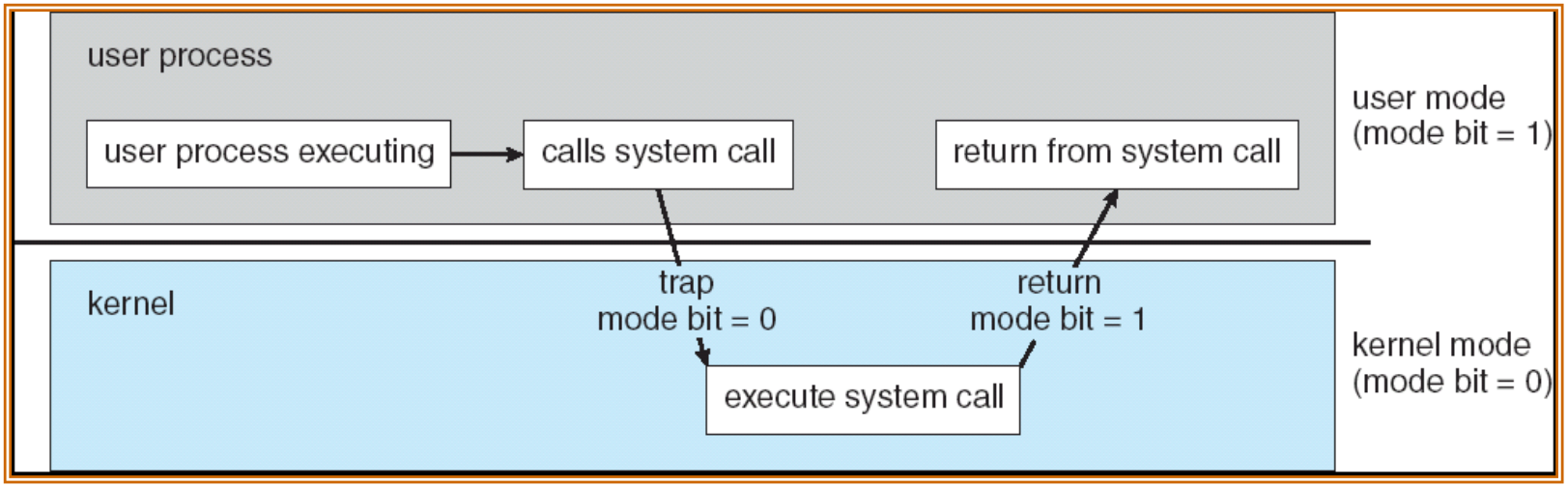
Ano letivo 2024/2025

Nuno Lau (nunolau@ua.pt)

Modos de operação

- De modo a garantir a segurança do sistema, a maioria dos SOs podem executar em 2 modos:
  - Modo de utilizador
    - Com restrições de segurança
    - Acesso a certas instruções e zonas de memória e dispositivos estão interditos
  - Modo de *kernel*
    - Sem restrições de segurança
    - Pode executar todas as instruções e acessos
    - Instruções privilegiadas
  - Chamadas ao sistemas providenciam uma forma segura de alternar entre os 2 modos

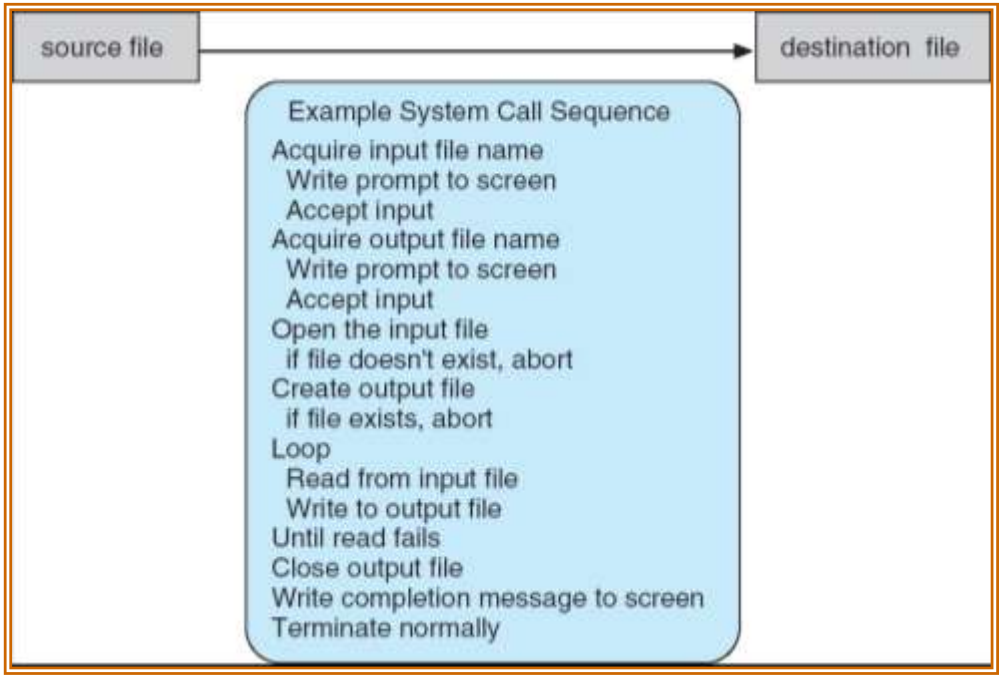
Modos de operação



Chamadas ao Sistema

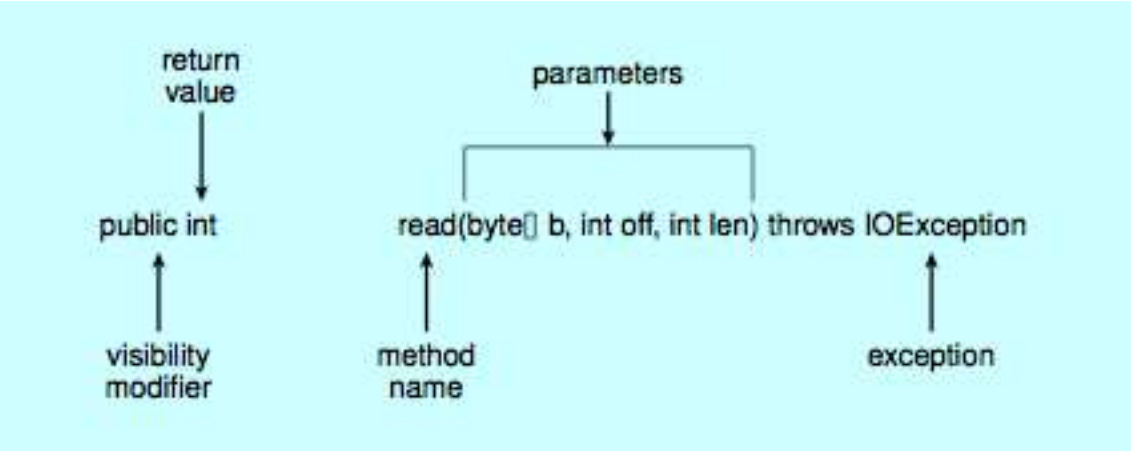
- Interface para acesso aos serviços do SO
- Tipicamente escrita numa linguagem de alto nível
- Programas usam, em geral, uma API para acesso a *system calls* em vez de utilização direta
- As 3 APIs mais comuns são:
  - **Win64 (Win32) API** para Windows
  - **POSIX API** para sistemas baseados em POSIX (UNIXs, Mac OS X)
  - **Java API** para a *Java Virtual Machine*
- Qual a razão de usar API em vez de usar a *system call* diretamente?

Chamadas ao Sistema

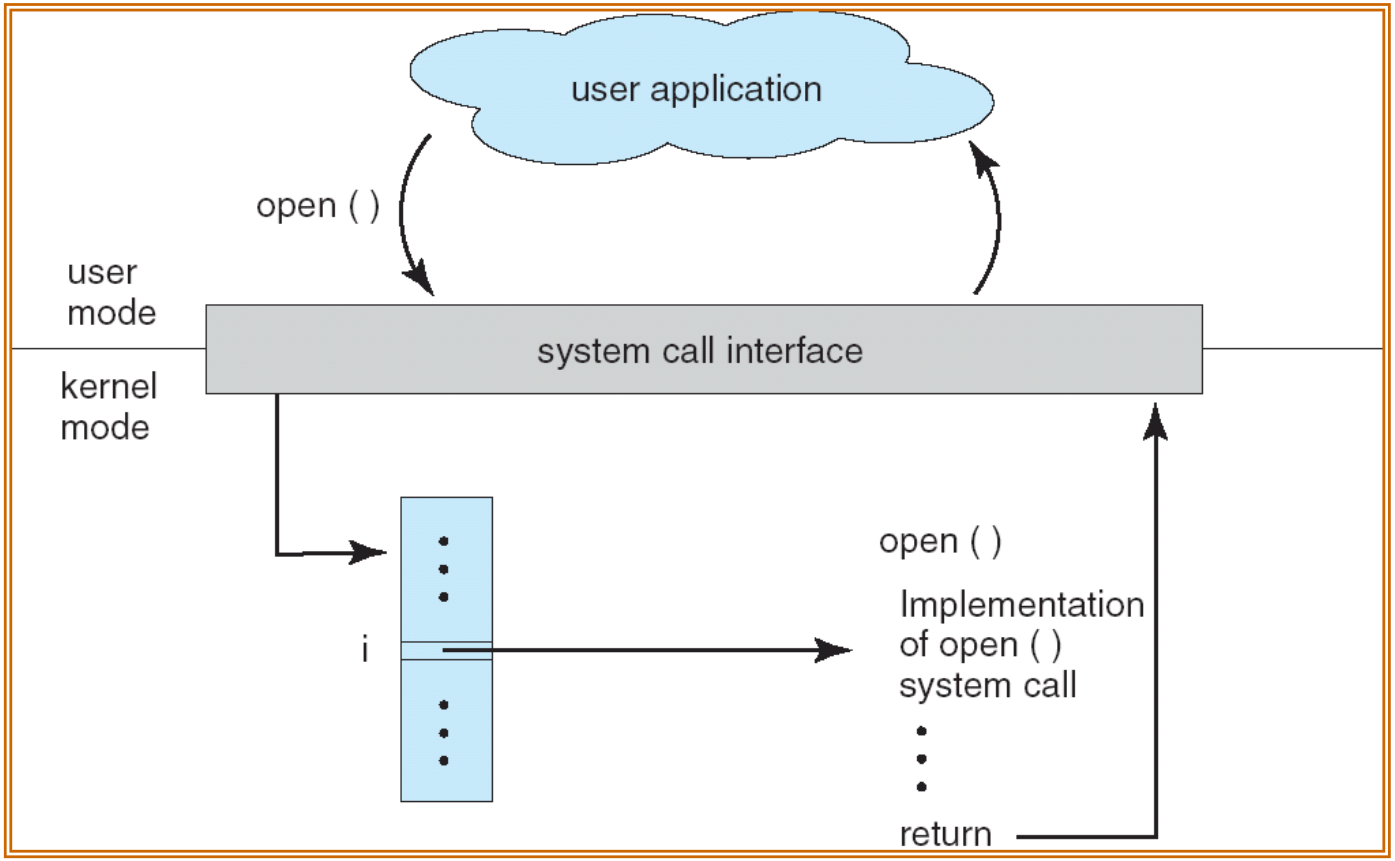


Chamadas ao Sistema

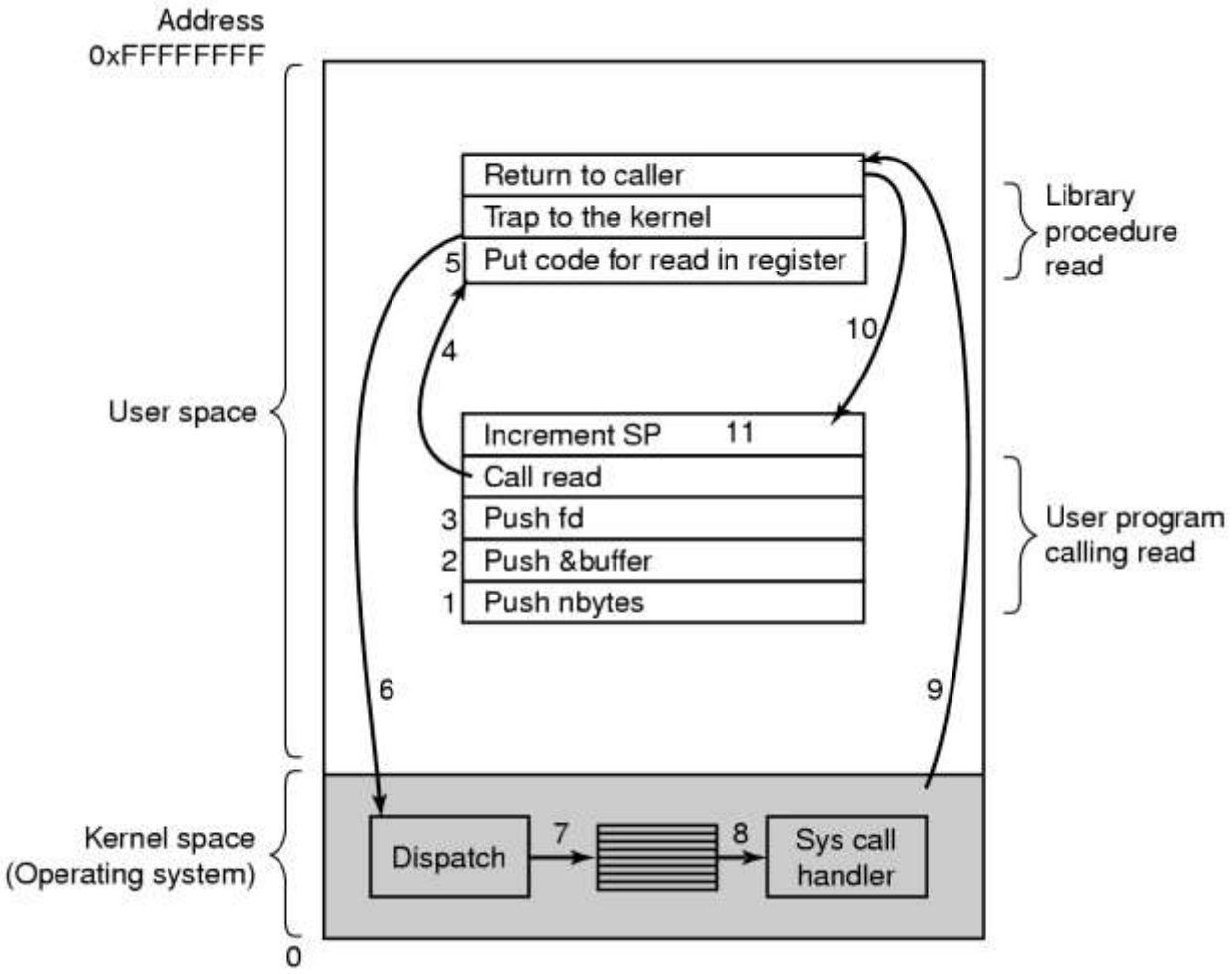
- Exemplo: Java read()



Chamadas ao Sistema



Chamadas ao Sistema



Chamadas ao Sistema

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Chamadas ao Sistema

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

Chamadas ao Sistema

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Chamadas ao Sistema

Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Chamadas ao Sistema

Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Interrupções/Excepções

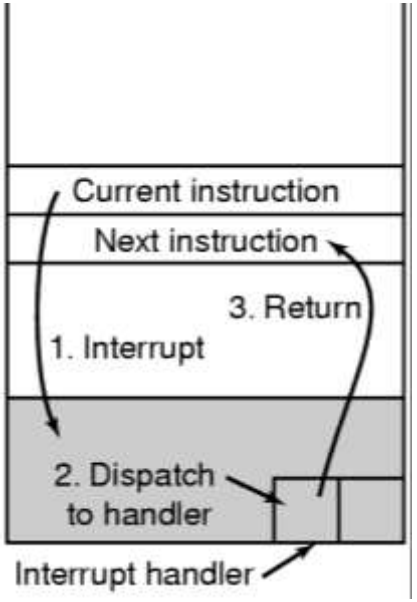
- Considerar um computador com apenas 1 CPU que está a executar o seguinte código:

```
while(1) {
        i++;
}
```

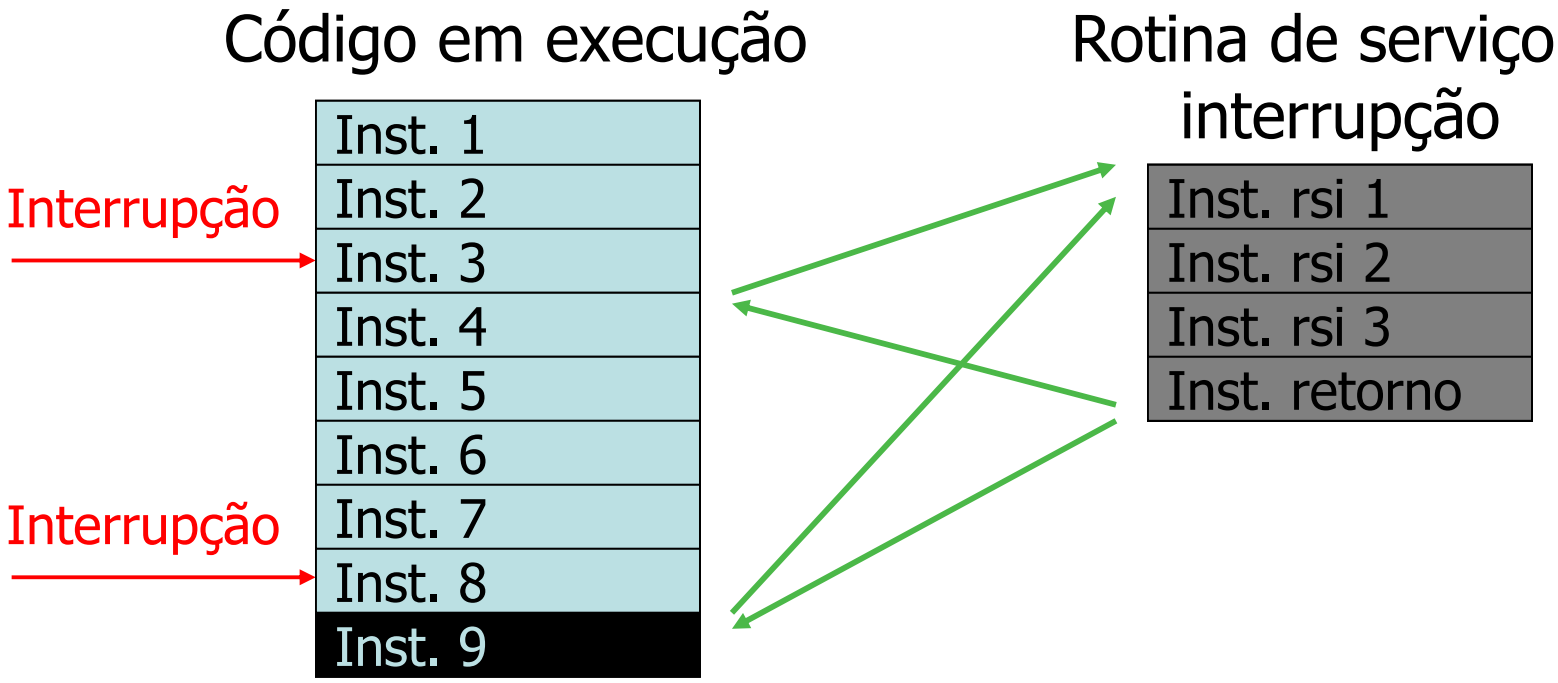
- Como pode o Sistema Operativo obter o controlo do computador?

Interrupções/Excepções

- Quando o CPU deteta uma interrupção **abandona o código** que está a executar e **transfere o controlo** para a **rotina de atendimento da interrupção**
- O endereço da instrução interrompida deve ser salvaguardado
  - Porquê?
- Durante a execução da rotina de atendimento as interrupções estão desativadas
  - Problema da interrupção perdida
- Um *trap* ou exceção é uma interrupção gerada por software
  - *Access violation*, *breakpoint*, *misaligned access*, *divide by 0*, *overflow*, *illegal instruction*, *previleged instruction*
- Nos Sistemas operativos as interrupções são fundamentais

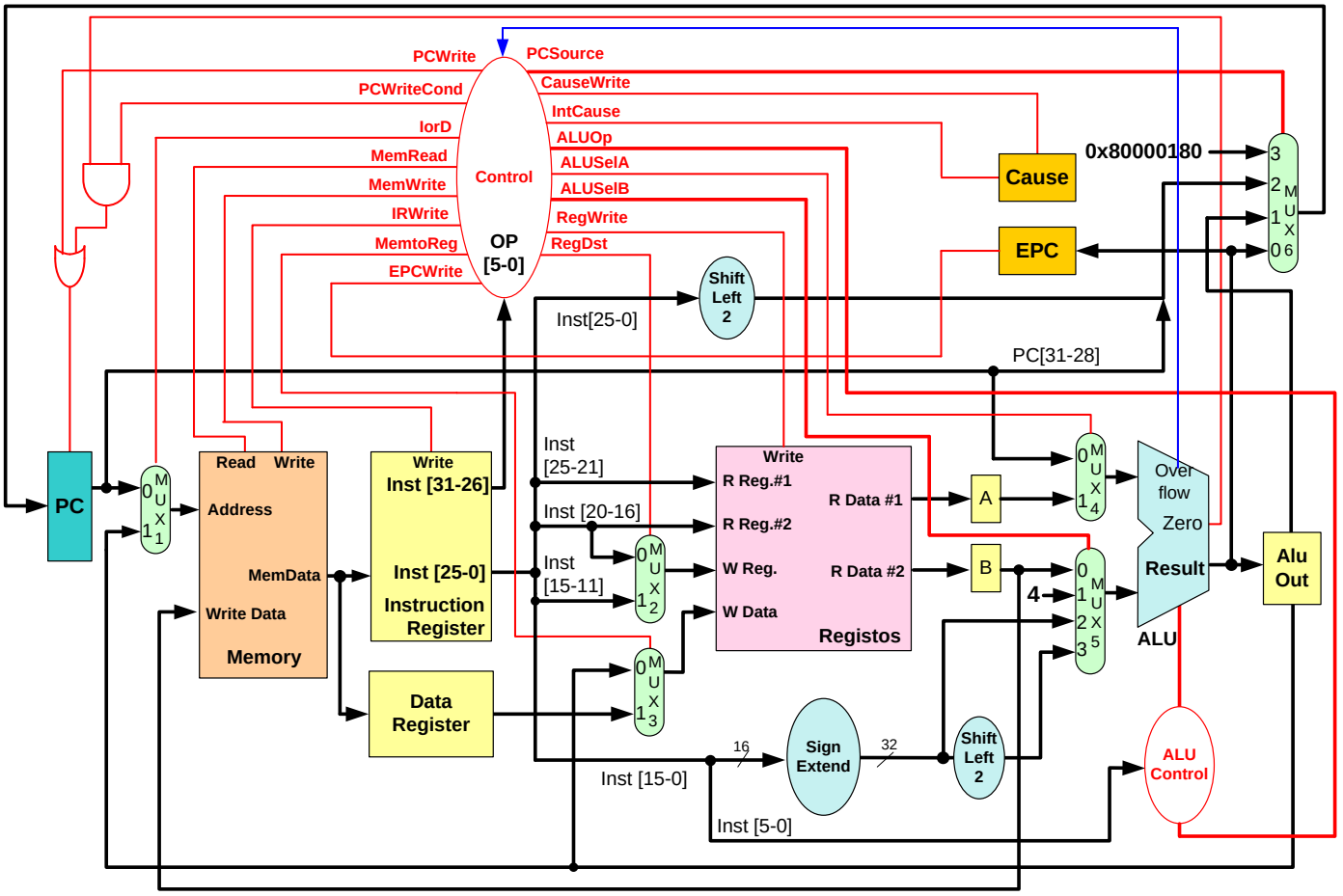


Interrupções/Excepções

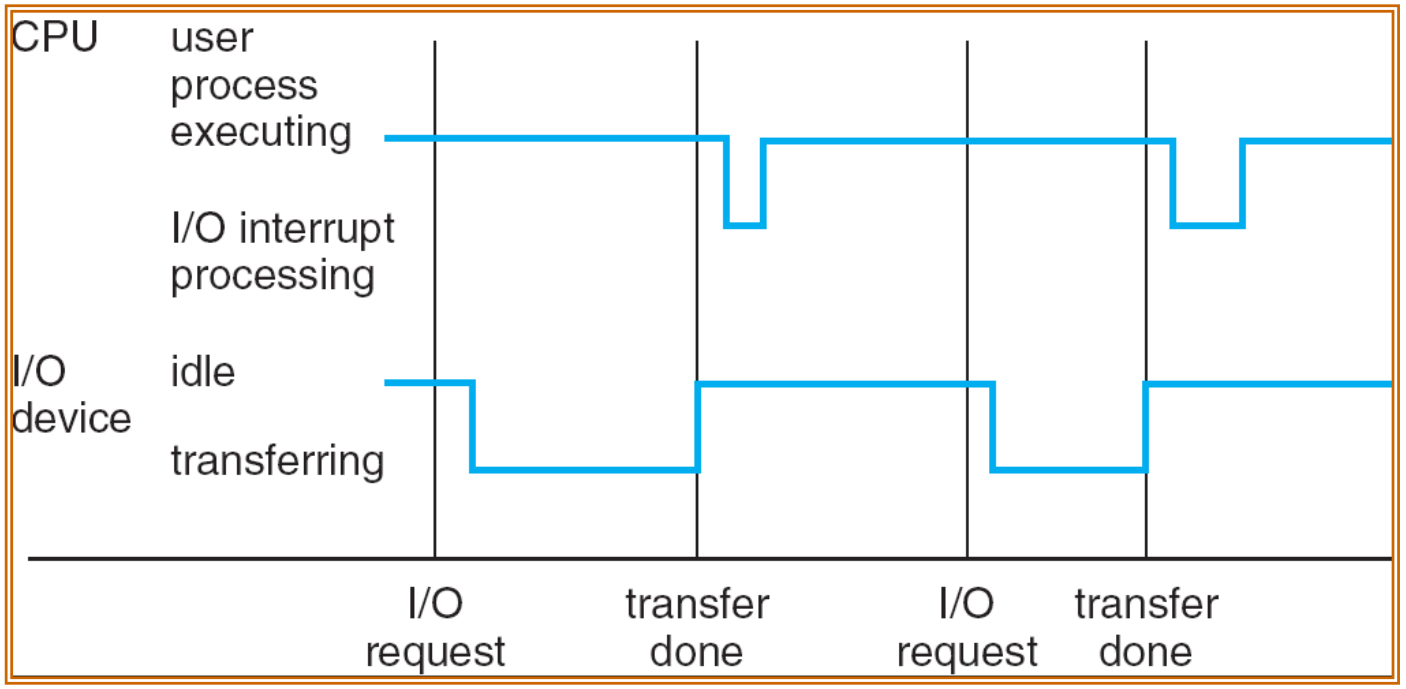


Interrupções/Excepções no MIPS

- Registos **Cause** e **EPS**
- Salto para **0x80000180** se **PCSource**=3



Atendimento de uma interrupção



- Dispositivo de I/O envia interrupção ao CPU quando "*transfer done*"
- CPU executa rotina de atendimento à interrupção, no âmbito do SO, e volta a executar processo do utilizador

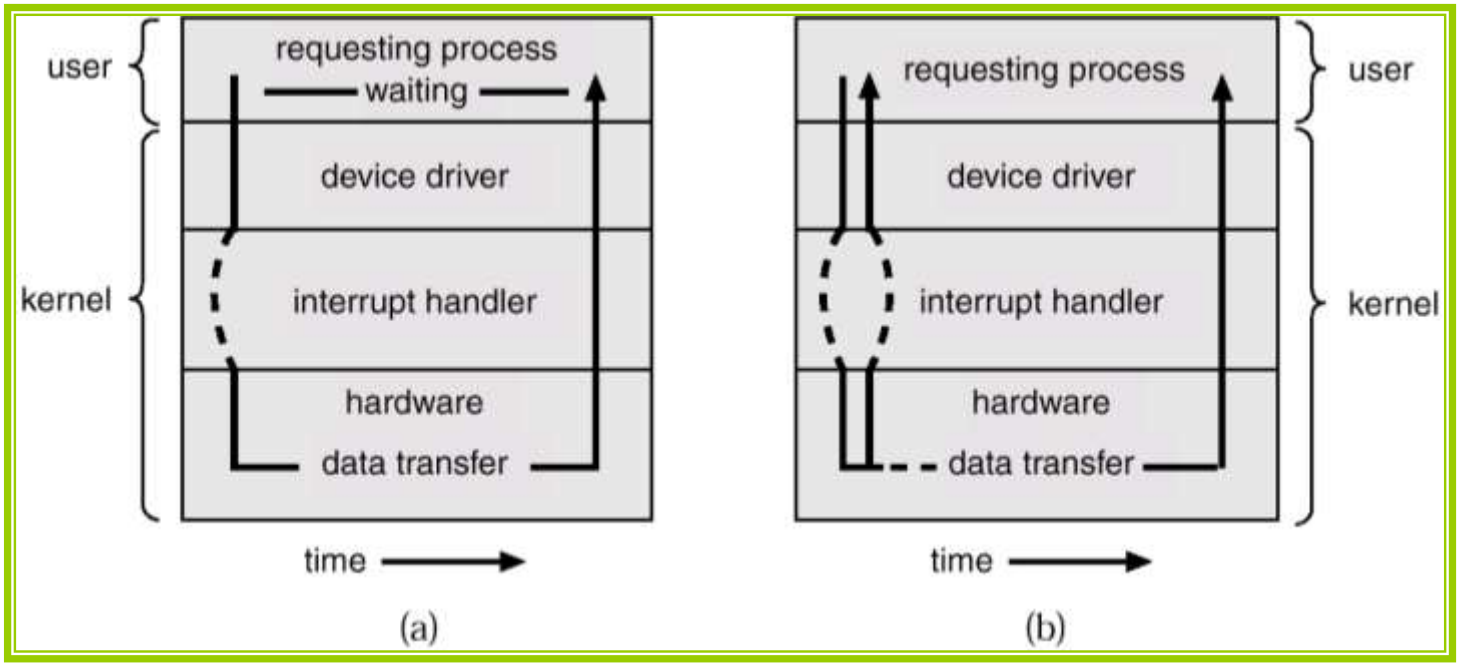
Tipos de Interrupções

- Interrupções de I/O
  - O dispositivo de I/O pede atenção. A rotina de atendimento deve aceder ao dispositivo para determinar a ação necessária.
  - Teclado, rato, placa de rede, disco, etc.
- Interrupções de *timers*
  - Dizem ao processador que um certo intervalo de tempo já decorreu.
  - Timer local ou timer externo
- Interrupções entre processadores
  - Um processador emite uma interrupção para outro processador num sistema multi-processador

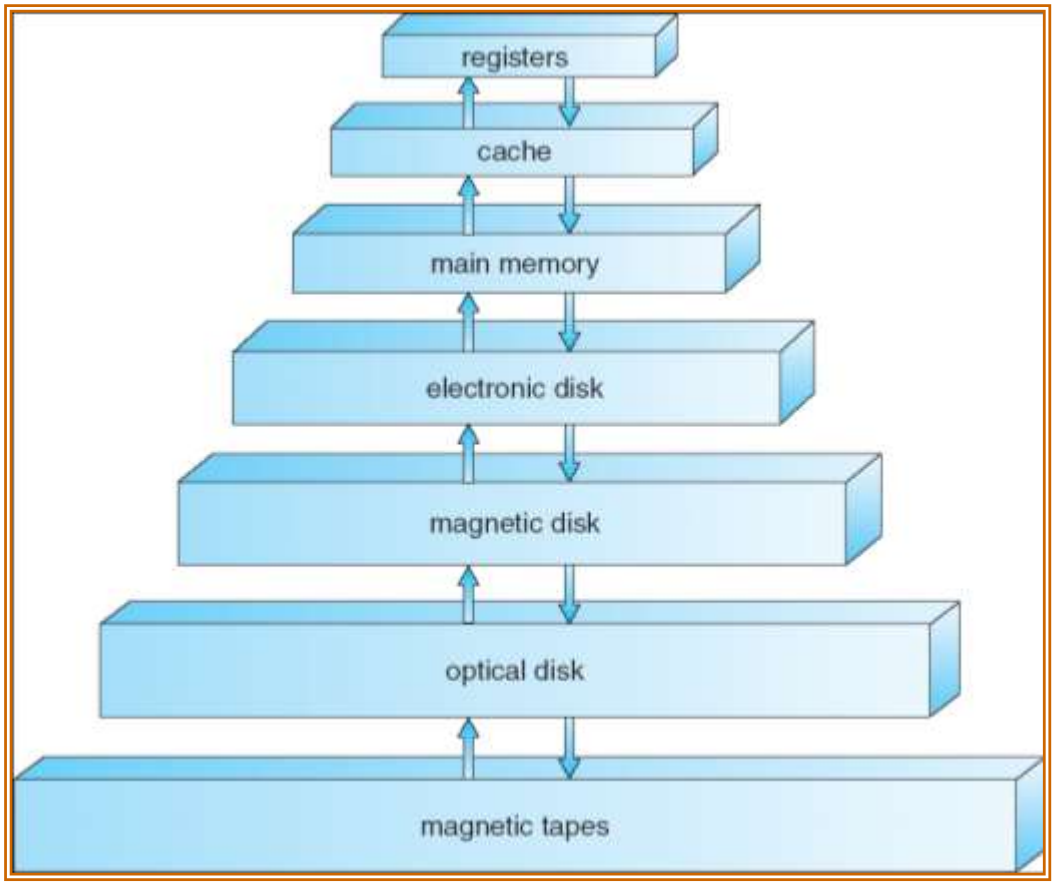
Organização de I/O

Síncrona

Assíncrona



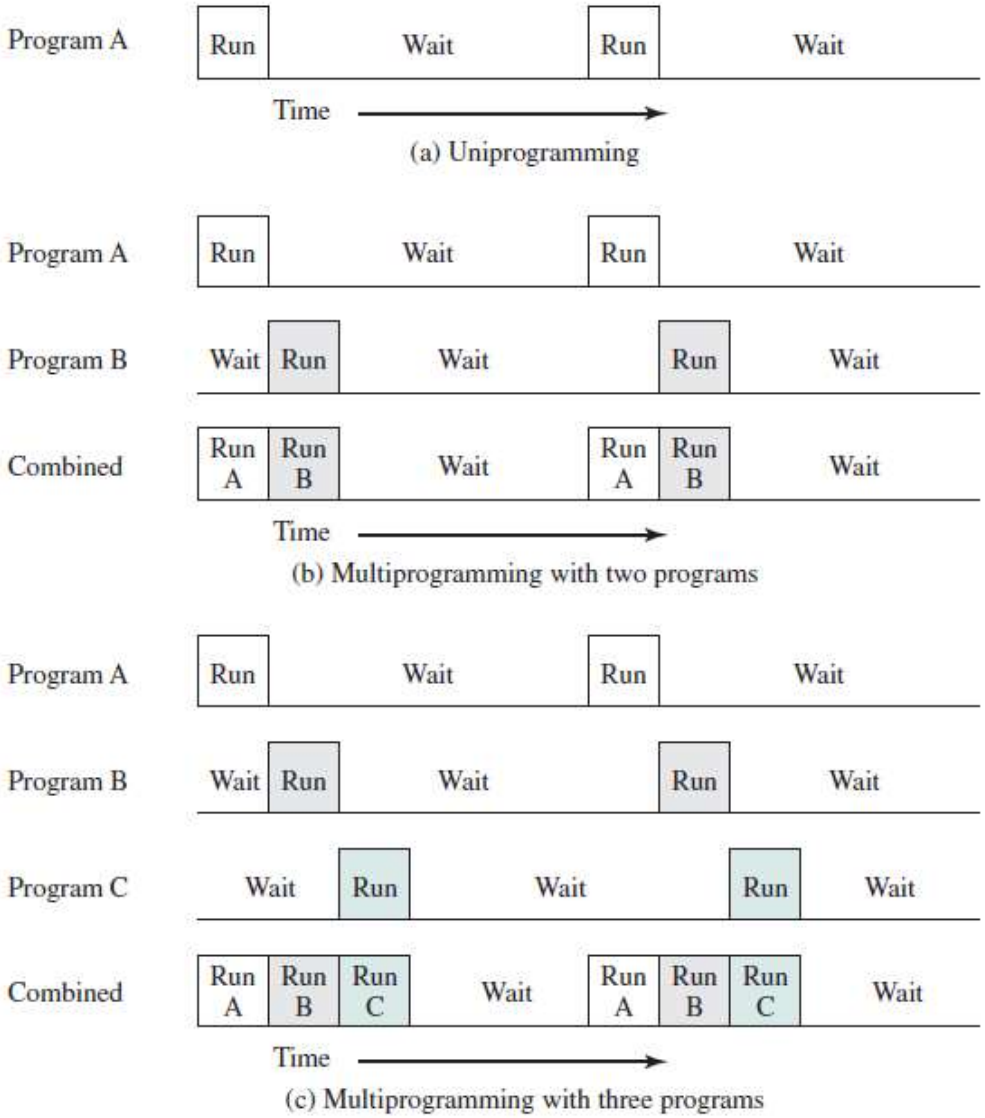
Hierarquia de armazenamento



Multiprogramação

- Permite que mais do que 1 processo (programa em execução) esteja ativo em simultâneo
- A utilização de apenas 1 processo não permite manter o CPU constantemente ativo
- Multiprogramação escolhe um novo processo para ocupar o CPU quando o processo que estava a ser executado tem de esperar (ex: chamada de I/O)
- Os processos ativos são mantidos em memória
- Escalonamento de processos

Multiprogramação



Timesharing

- CPU altera o processo em execução mesmo que este não necessite de esperar
- A cada processo é atribuído um tempo máximo de ocupação consecutiva do processador
- Se esse tempo é esgotado o SO **muda o contexto** do processador para outro processo
- Diminui muito o tempo de resposta de aplicações interactivas
- Permite que vários utilizadores usem o mesmo sistema computacional como se dispusessem do sistema em exclusivo

Lista de interrupções

- Para visualizar configuração de interrupções
  - Windows
    - Aplicação **`MSInfo32.exe`**
      - Hardware Resources -> IRQs
  - Linux
    - Ficheiro **`/proc/interrupts`**