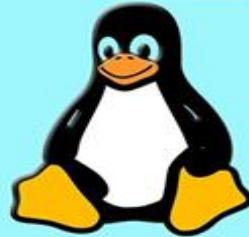


SISTEMAS OPERATIVOS



ANDROID



INFORMÁTICA DE GESTÃO

Professor Doutor Ricardo Vardasca, PhD, ASIS, FRPS
ricardo.vardasca@islasantarem.pt

ISLA

Santarém

02 - Introdução aos Sistemas Operativos II

- Introdução aos Sistemas Operativos:
 - Definição
 - Objectivos
 - Camadas
 - Serviços fornecidos
 - Conceito de kernel
- Evolução dos sistemas operativos

- Arquitectura do S.O.:
 - Elementos chave
 - Estrutura do S.O.
 - Decomposição funcional
 - Micro-núcleo
 - Estrutura monolítica
- Conceitos presentes em S.O. modernos
- Abstrações suportadas por um S.O.
- Necessidades de suporte e de proteção de hardware

Elementos chave presentes em qualquer S.O.:

- Processo
- Gestão de memória
- Segurança e protecção da informação
- Gestão e escalonamento de recursos

Pilar na estrutura do sistema operativo

- Termo mais generalizado para uma tarefa (JOB)

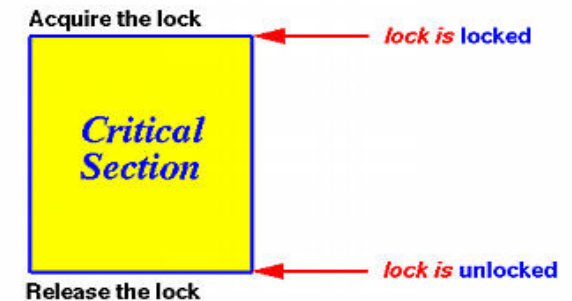
- Várias definições possíveis:

- Um programa em execução
- A entidade que pode ser atribuída ou executada num processador
- Uma actividade caracterizada por um único fluxo de execução, um estado corrente e um conjunto de recursos do sistema associado

Process	PID	CPU	Description
System Idle Process	0	88.93	
Interrupts	n/a	0.38	Hardware Interrupts
DPCs	n/a	0.76	Deferred Procedure Calls
System	4		
SMSS.EXE	704		Windows NT Session Manager
CSRSS.EXE	896		Client Server Runtime Process
WINLOGON.EXE	1072		Windows NT Logon Application
SERVICES.EXE	1116	0.76	Services and Controller app
SVCHOST.EXE	1300		Generic Host Process for Win32 Services

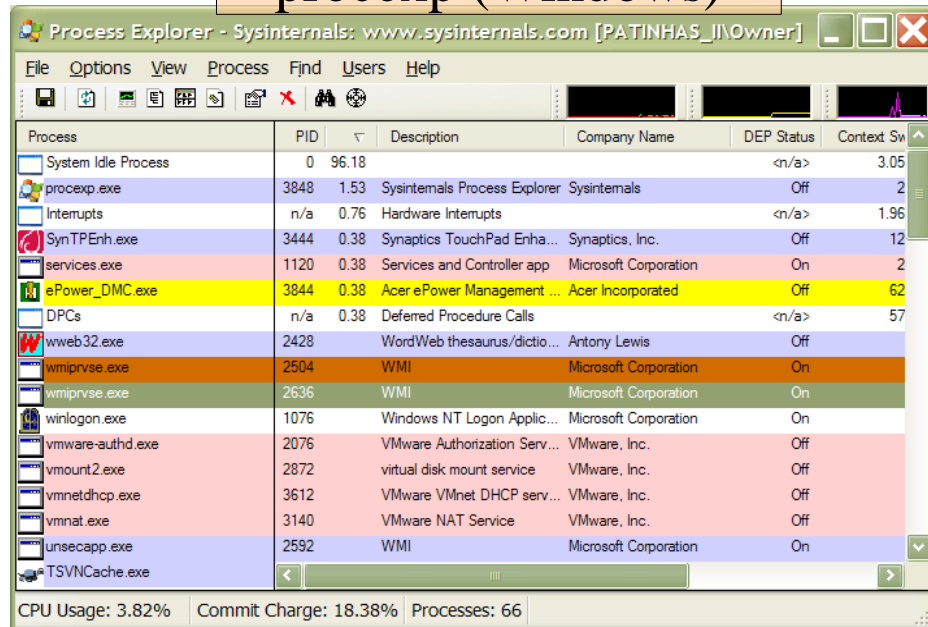


- O sistema operativo necessita de garantir uma correcta sincronização entre processos
 - Assegurar exclusão mútua no acesso a recursos
 - Evitar situações de *deadlock*
- Dificuldades encontradas:
 - Sincronização
 - Assegurar que um processo à espera de um dispositivo de E/S receba o sinal quando este estiver disponível
 - Exclusão mútua
 - Evitar que dois processos acessem ao mesmo tempo a um determinado recurso
 - Ex: Atribuição do mesmo lugar a 2 passageiros



Lista de processos

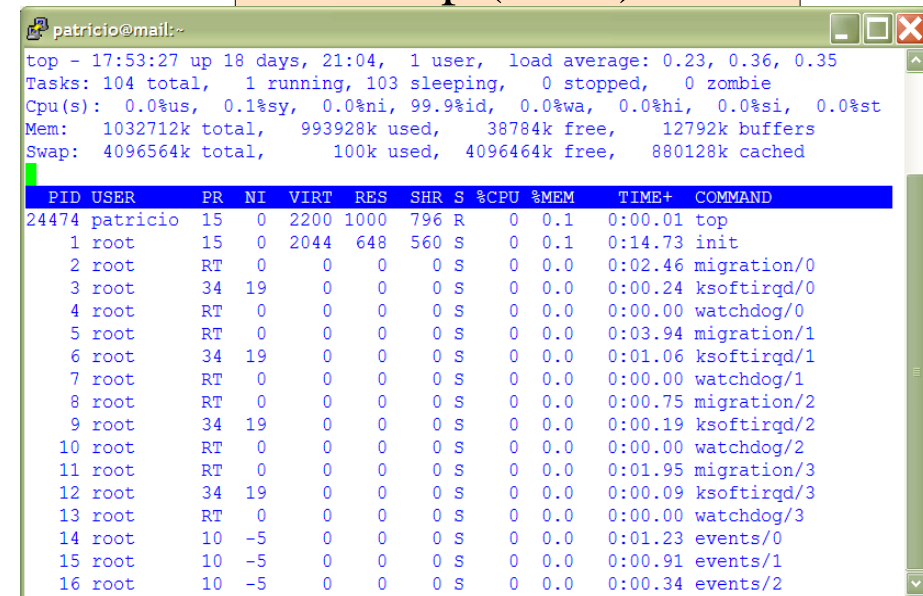
procexp (Windows)



Process	PID	Description	Company Name	DEP Status	Context Sw
System Idle Process	0	96.18		<n/a>	3.05
procexp.exe	3848	1.53	Sysinternals Process Explorer Sysinternals	Off	2
Interrupts	n/a	0.76	Hardware Interrupts	<n/a>	1.96
SynTPEnh.exe	3444	0.38	Synaptics TouchPad Enh...	Off	12
services.exe	1120	0.38	Services and Controller app	On	2
ePower_DMC.exe	3844	0.38	Acer ePower Management ...	Off	62
DPCs	n/a	0.38	Deferred Procedure Calls	<n/a>	57
wwweb32.exe	2428	WordWeb thesaurus/dictio...	Antony Lewis	Off	
wmiprvse.exe	2504	WMI	Microsoft Corporation	On	
wmiprvse.exe	2636	WMI	Microsoft Corporation	On	
winlogon.exe	1076	Windows NT Logon Applic...	Microsoft Corporation	On	
vmware-authd.exe	2076	VMware Authorization Serv...	VMware, Inc.	Off	
vmount2.exe	2872	virtual disk mount service	VMware, Inc.	Off	
vmnetdhcp.exe	3612	VMware VMnet DHCP serv...	VMware, Inc.	Off	
vmnat.exe	3140	VMware NAT Service	VMware, Inc.	Off	
unsecapp.exe	2592	WMI	Microsoft Corporation	On	
TSVNCache.exe					

CPU Usage: 3.82% Commit Charge: 18.38% Processes: 66

top (linux)



```
top - 17:53:27 up 18 days, 21:04, 1 user, load average: 0.23, 0.36, 0.35
Tasks: 104 total, 1 running, 103 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.1%sy, 0.0%ni, 99.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1032712k total, 993928k used, 38784k free, 12792k buffers
Swap: 4096564k total, 100k used, 4096464k free, 880128k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
24474	patricio	15	0	2200	1000	796	R	0	0.1	0:00.01	top
1	root	15	0	2044	648	560	S	0	0.1	0:14.73	init
2	root	RT	0	0	0	0	S	0	0.0	0:02.46	migration/0
3	root	34	19	0	0	0	S	0	0.0	0:00.24	ksoftirqd/0
4	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/0
5	root	RT	0	0	0	0	S	0	0.0	0:03.94	migration/1
6	root	34	19	0	0	0	S	0	0.0	0:01.06	ksoftirqd/1
7	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/1
8	root	RT	0	0	0	0	S	0	0.0	0:00.75	migration/2
9	root	34	19	0	0	0	S	0	0.0	0:00.19	ksoftirqd/2
10	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/2
11	root	RT	0	0	0	0	S	0	0.0	0:01.95	migration/3
12	root	34	19	0	0	0	S	0	0.0	0:00.09	ksoftirqd/3
13	root	RT	0	0	0	0	S	0	0.0	0:00.00	watchdog/3
14	root	10	-5	0	0	0	S	0	0.0	0:01.23	events/0
15	root	10	-5	0	0	0	S	0	0.0	0:00.91	events/1
16	root	10	-5	0	0	0	S	0	0.0	0:00.34	events/2

- Dificuldades encontradas (cont.):
 - Execução do programa não é determinística
 - A execução de um programa apenas deve depender dos seus dados de entrada e não das actividades dos restantes processos

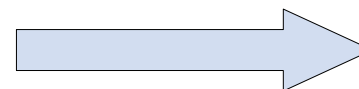
Execução não determinística

```
1: int counter = 0;
2: void thread_start(){
3:   int local;
4:   lock();
5:   counter = counter + thread_id();
6:   local = counter;
7:   unlock();
8:   printf("%d\n", local);
9: }
10: void main(){
11:   thread_create(thread_start); // thread id = 1
12:   thread_create(thread_start); // thread id = 2
13:   thread_create(thread_start); // thread id = 3
14: }
```

Replica 1:	Replica 2:
Thread 1 prints "1"	Thread 2 prints "2"
Thread 3 prints "4"	Thread 3 prints "5"
Thread 2 prints "6"	Thread 1 prints "6"

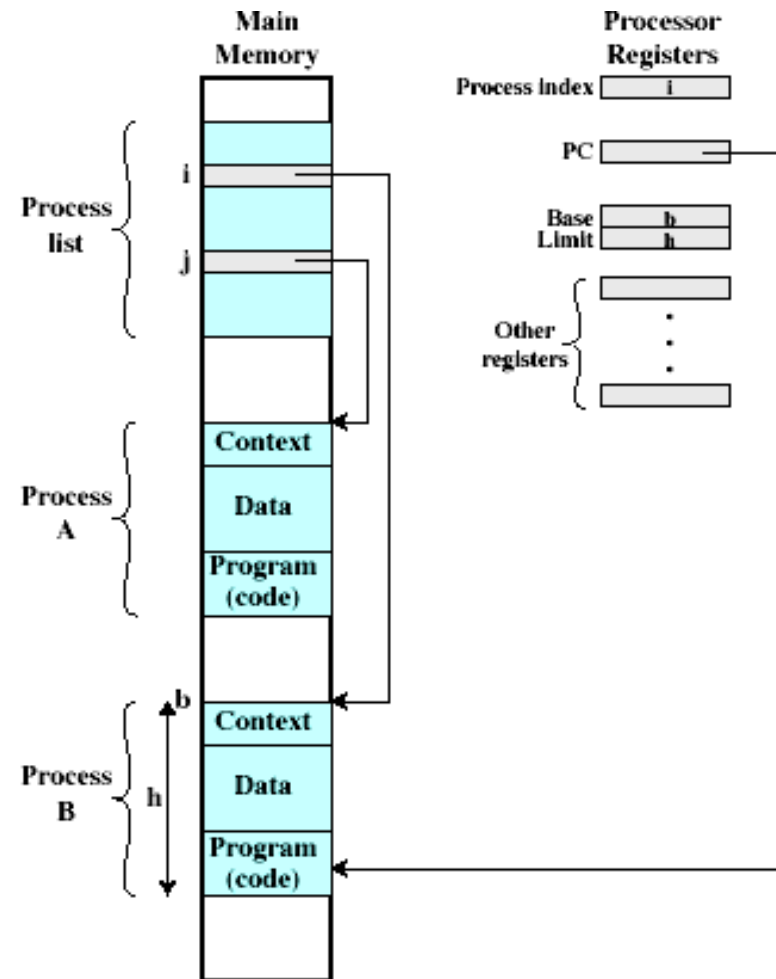
- *Deadlocks*

- Situação em que dois (ou mais) processos ficam bloqueados à espera um do outro
 - Deste modo, está-se num impasse: os processos não progridem



Um processo é constituído por três componentes básicos:

- Executável do programa (e.g., ficheiro EXE no Windows)
- Os dados necessários à execução de um programa (variáveis, dados de entrada, buffers, etc)
- O contexto de execução do programa
 - Toda a informação que o sistema operativo requer para gerir o processo. Por exemplo:
 - Conteúdo dos registos (registos de dados, PC, etc)
 - PC: program counter
 - Prioridade do processo
 - Se está à espera de um evento de E/S
 - etc



O estado de um processo é plenamente representado pelo seu contexto

Listagem de processos

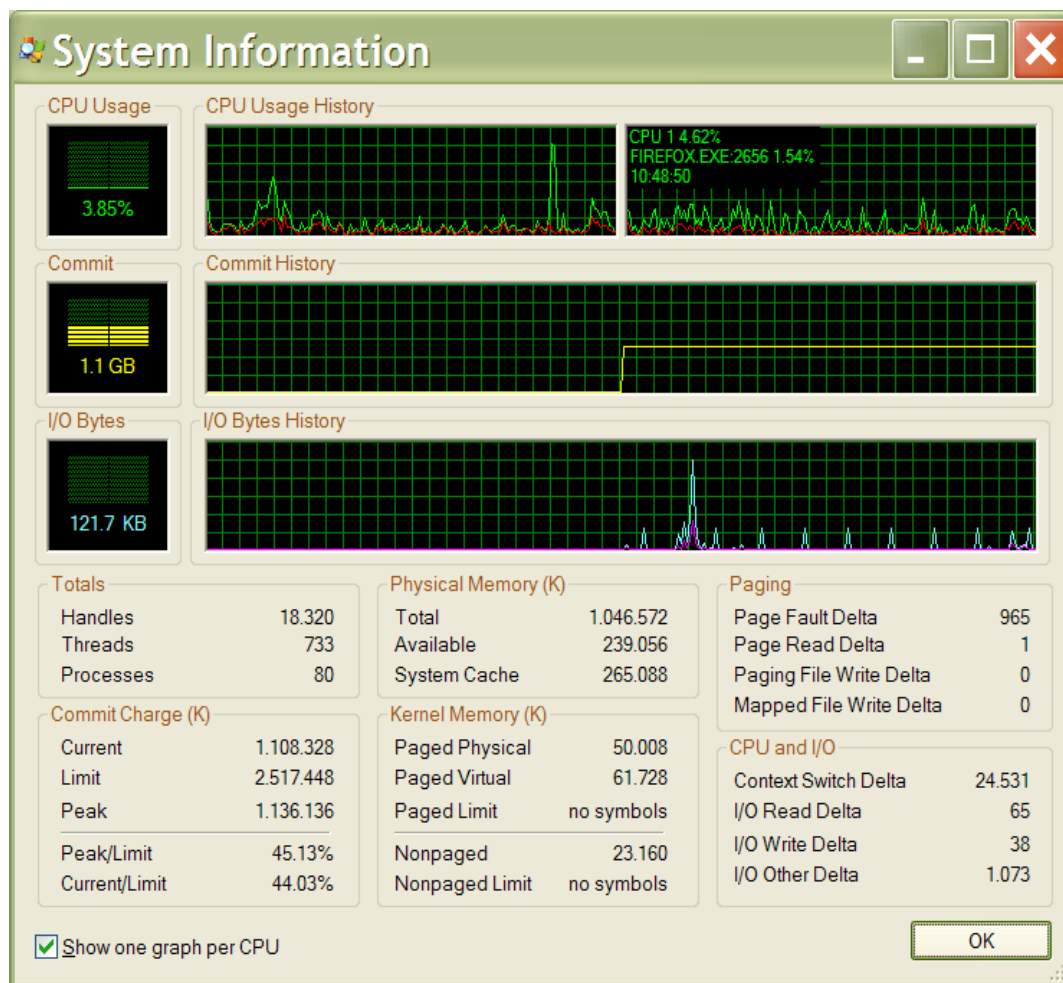
Process	PID	CPU	Description	Company Name
SQLSERVER.EXE	1904		SQL Server Windows NT	Microsoft Corporati
sqlbrowser.exe	2268		SQL Browser Service EXE	Microsoft Corporati
SPOOLSV.EXE	932		Spooler SubSystem App	Microsoft Corporati
soffice.exe	3620		OpenOffice.org 3.2	OpenOffice.org
soffice.bin	3624		OpenOffice.org 3.2	OpenOffice.org
SMSS.EXE	912		Windows NT Session Manager	Microsoft Corporati
SERVICES.EXE	1612	0.38	Services and Controller app	Microsoft Corporati
S24EvMon.exe	1388		Wireless Management Service	Intel Corporation
RUNDLL32.EXE	1012		Run a DLL as an App	Microsoft Corporati
RUNDLL32.EXE	3980	0.38	Run a DLL as an App	Microsoft Corporati
RegSrv.exe	2152		Intel(R) PROSet/Wireless Registry Service	Intel Corporation
procexp.exe	3132	0.38	Sysinternals Process Explorer	Sysinternals
POWERPNT.EXE	768		Microsoft Office PowerPoint	Microsoft Corporati
NVSVC32.EXE	2120		NVIDIA Driver Helper Service, Version 86.02	NVIDIA Corporation
mspaint.exe	4004		Paint	Microsoft Corporati
Monitor.exe	3928		Monitor	acer Inc.
MCRDSVC.EXE	2360		MCRD Device Service	Microsoft Corporati
LSSvc.exe	3984			Hewlett-Packard C
LSASS.EXE	1624		LSA Shell (Export Version)	Microsoft Corporati
LManager.exe				
Launchv.exe				

CPU Usage: 2.69% Commit Charge: 44.43% Processes: 80

Process Explorer:
processos
existentes
numa máquina
core 2 duo

www.sysinternals.com - procexp

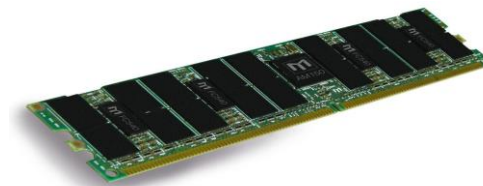
A utilização dos recursos do sistema



procexp (Windows)

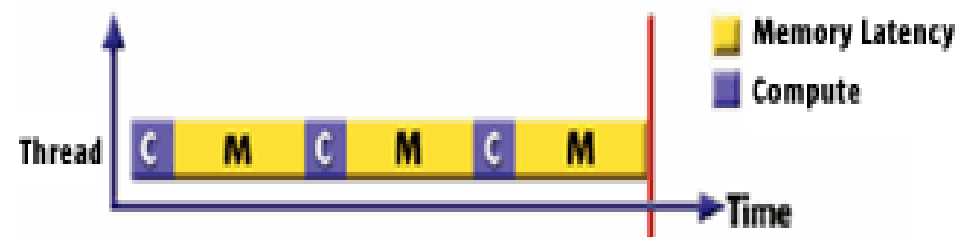
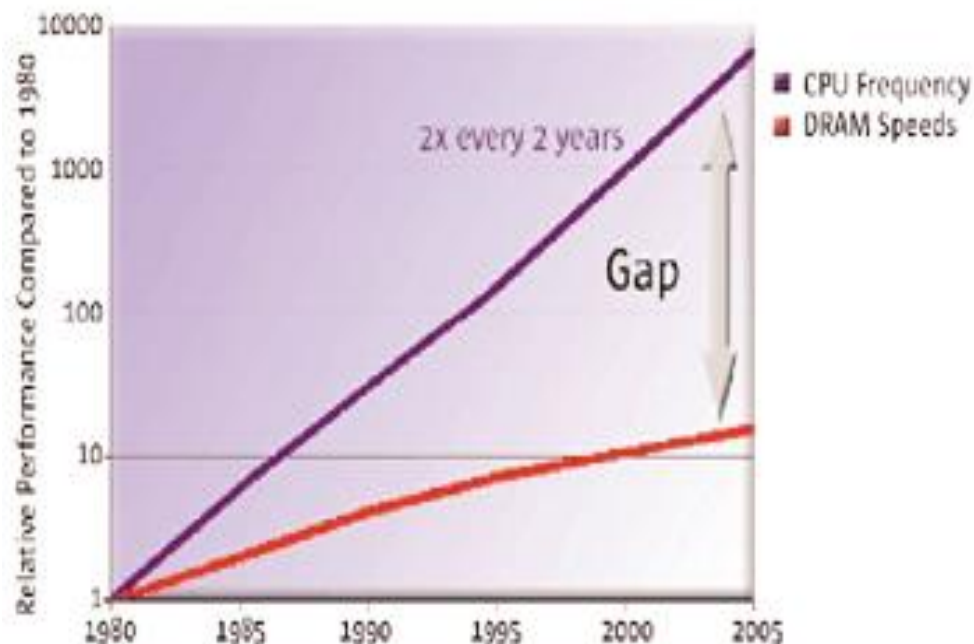
Responsabilidades do sistema operativo:

- Isolamento dos processos
 - Evitar que os processos interfiram com a memória um do outro (dados e instruções)
- Gestão e alocação automática
 - A alocação de memória deve ser totalmente transparente para o programador (uso de malloc, free, etc.)
- Suporte para programação modular
 - Deve permitir a um programa constituído por módulos, carregar, destruir ou alterar dinamicamente os mesmos
- Protecção e acesso controlado a zonas de memória partilhadas
- Armazenamento a longo prazo (ex.: hibernate)



Velocidade CPU vs velocidade RAM

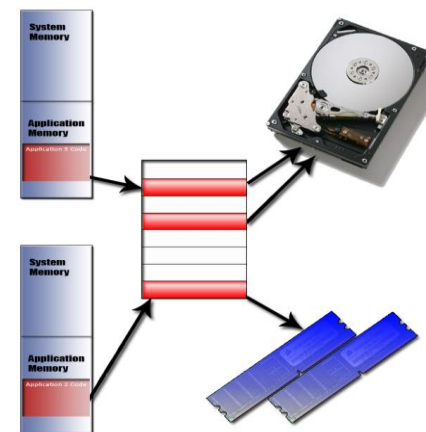
- Não esquecer: o fosso de velocidade entre CPU e memória RAM continua a crescer!
- 75% do tempo de CPU é tipicamente perdido...à espera da memória

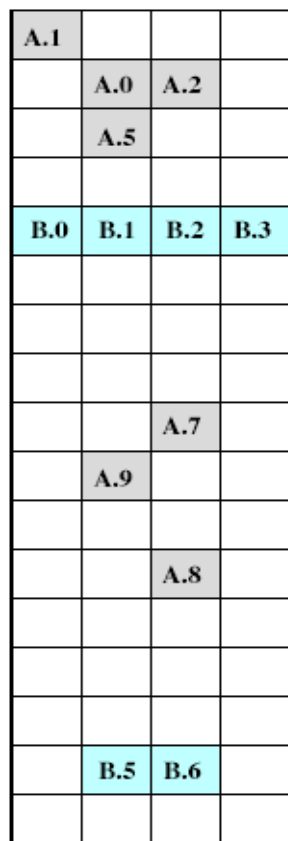


Fonte: <http://www.sun.com/processors/throughput/datasheet.html>

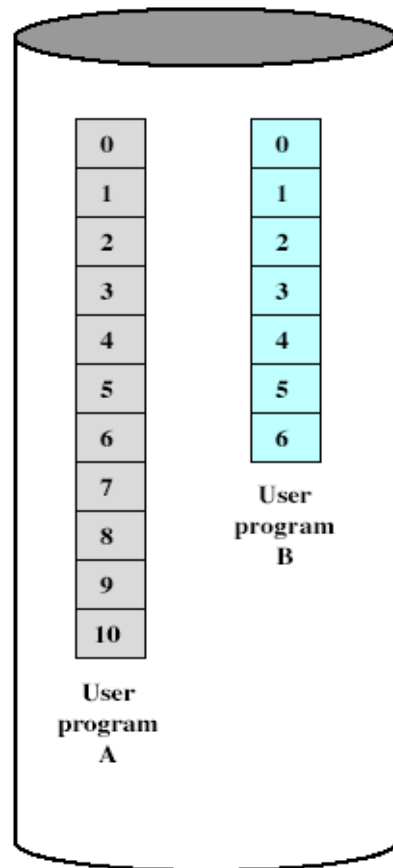
Memória virtual

- Permite a um programa referenciar a memória em termos lógicos independentemente da memória física disponível
- Mecanismo essencial à multi-programação para uma rápida comutação de processos
 - Como o tamanho de cada processo pode variar, a memória é dividida num conjunto de blocos de tamanho fixo (*paging*)
 - Um processo pode ocupar várias páginas
 - Um endereço de memória virtual é constituído por:
 - Número da página
 - *Offset* relativo à página





Main Memory



Disk

0 *paging* permite:

- Armazenamento a longo prazo
- Isolamento de processos
- Zonas de memória partilhada

Não é necessário conter todas as páginas de um programa em memória física

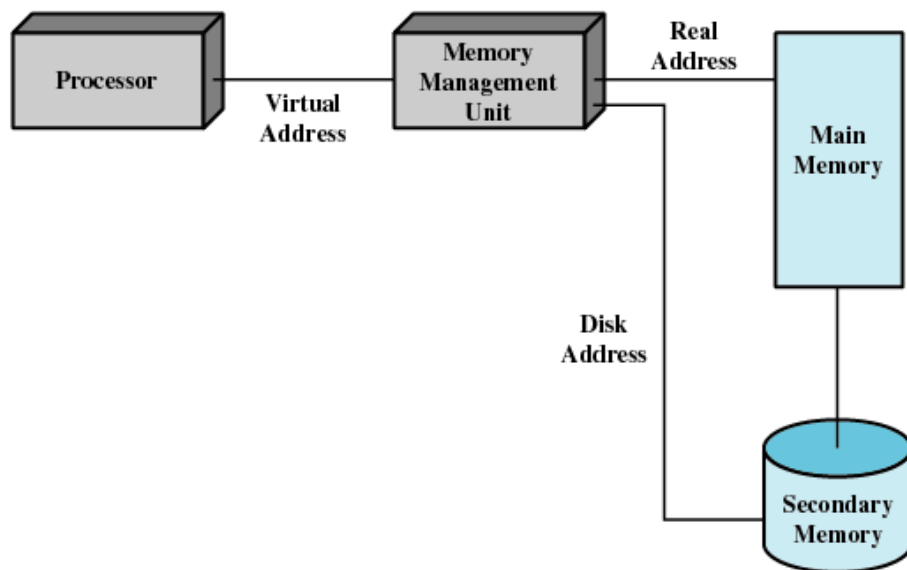
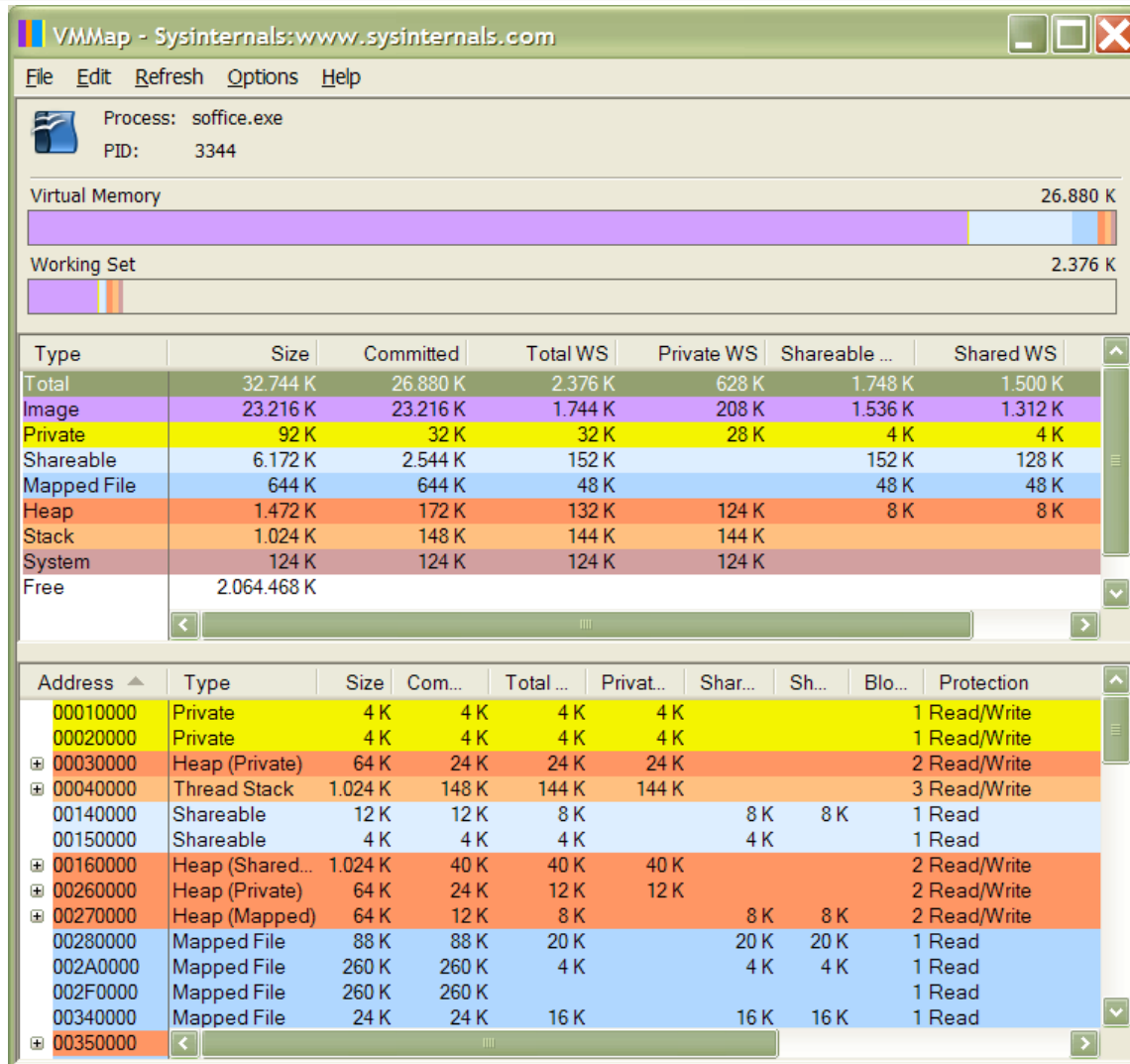


Figure 2.10 Virtual Memory Addressing

A unidade de gestão de memória é responsável por efectuar um mapeamento entre endereços virtuais e endereços reais

- Caso a página esteja em memória secundária, tem de ser carregada para a memória principal (suspendendo o processo)

Imagem de um processo em memória



VMMap – www.sysinternals.com

Os serviços de segurança e protecção da informação podem ser agrupados nas seguintes categorias:

- Disponibilidade
 - Proteger o sistema contra interrupções
- Confidencialidade
 - Assegurar que os utilizadores não têm acesso a informação para a qual não estão autorizados
- Integridade
 - Proteger os dados de alterações não autorizadas
- Autenticidade
 - Verificação da autenticidade de um utilizador e a validação de dados

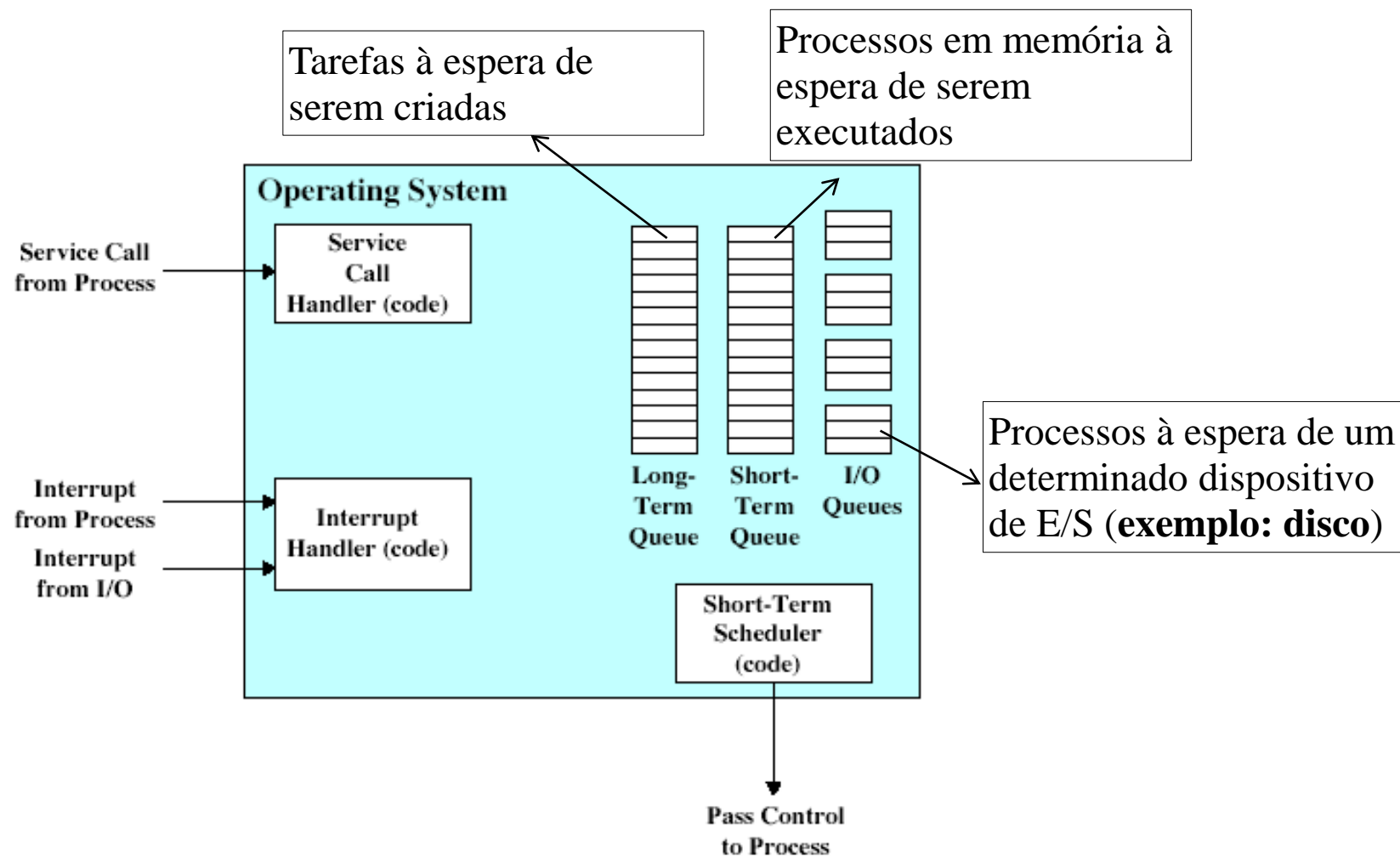


O S.O. deve gerir os dispositivos de E/S atendendo a três factores:

- Equidade
 - Tratamento igualitário para os processos que competem pelo mesmo recurso
- Resposta diferenciada
 - Diferenciar os processos de acordo com a classe a que pertencem e os recursos pretendidos
- Eficiência
 - Maximizar o débito, minimizar o tempo de resposta e acomodar o maior número de pedidos possível
- Objectivos contraditórios!



Principais elementos de um S.O.



Um S.O. é um sistema complexo

“The technique of mastering complexity has been known since ancient times: divide et impera (divide and rule)”

E. Dijkstra

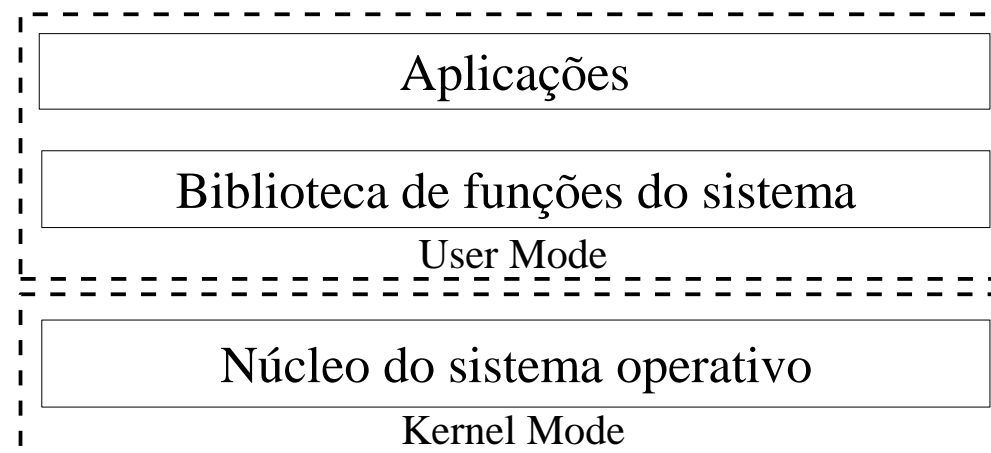
O S.O. é decomposto em camadas funcionais

- Cada camada é responsável por fornecer um determinado conjunto de serviços
- Cada camada depende da camada inferior para implementar e adicionar novos serviços

- A forma como as funcionalidades ou camadas funcionais do sistema operativo estão organizadas tem evoluído ao longo dos anos
- A maior parte dos sistemas operativos enquadra-se numa das seguintes estruturas:
 - Estrutura monolítica
 - Estrutura micro-núcleo (*micro-kernel*)

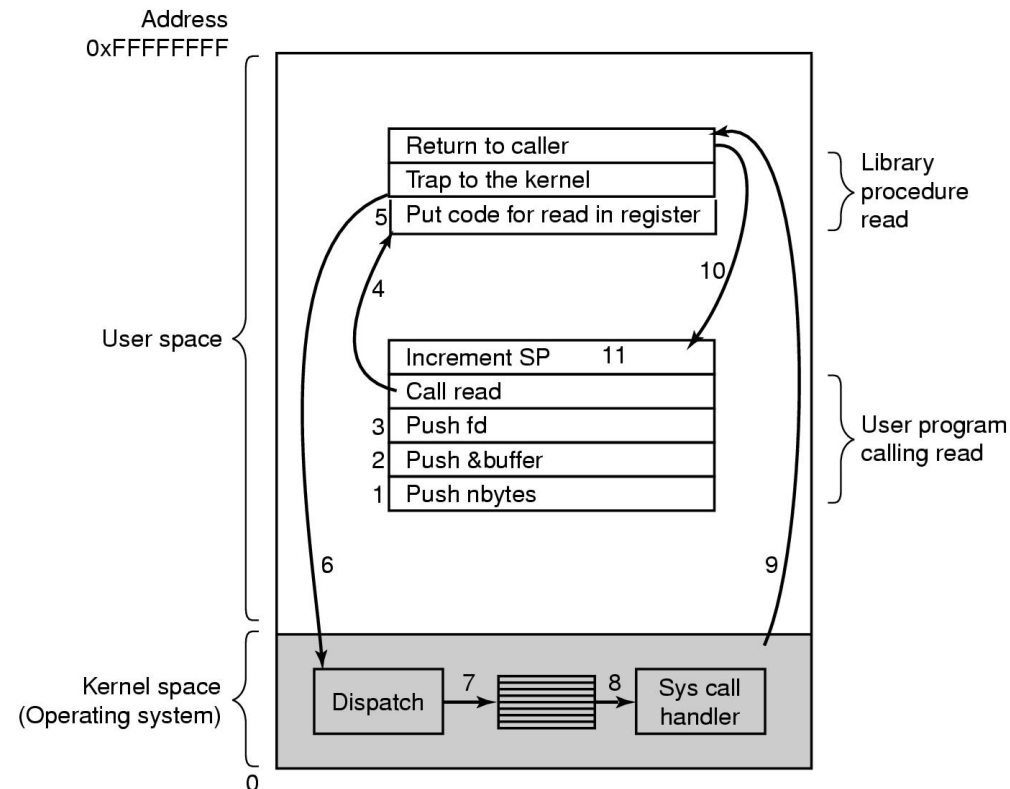
Estrutura monolítica

- O núcleo contém a maioria das funcionalidades do sistema operativo
- As funcionalidades disponibilizadas pelo S.O. são acedidas através da biblioteca de funções do sistema
- O espaço de endereçamento do *kernel* é separado do espaço de endereçamento do utilizador
- O código kernel corre todo num espaço de endereçamento único (melhor desempenho, pois evita mudanças de contexto)



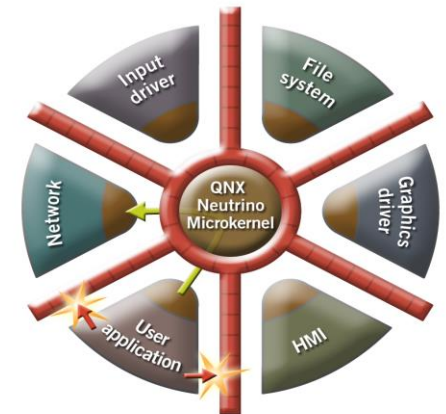
Estrutura monolítica

- Chamadas ao sistema (ex. `read(fd, buffer, nbytes)`)

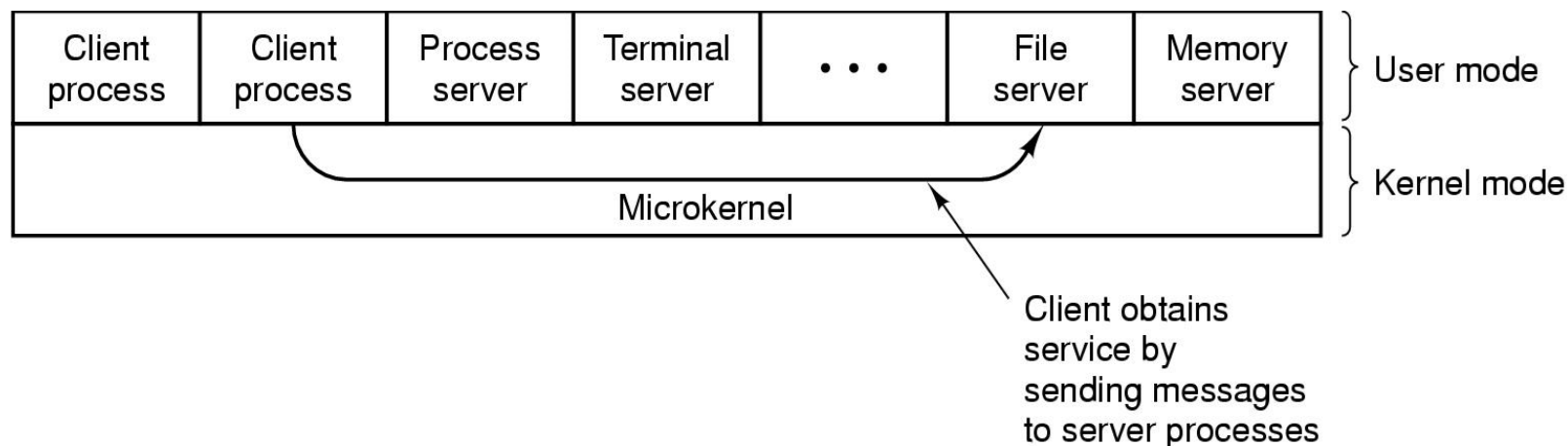


Estrutura de micro-kernel

- O *kernel* apenas contém as funções essenciais
 - Gestão dos espaços de endereçamento
 - Escalonamento de processos
 - Comunicação entre processos
 - Gestão de interrupções
- As restantes funcionalidades (gestão de processos, memória virtual, sistema de ficheiros, device drivers, etc) são disponibilizadas na forma de processos executados no modo utilizador
- No modo utilizador passam a existir 2 tipos de processos:
 - Processos cliente – processos do utilizador
 - Processos servidor – processos que providenciam um determinado serviço do S.O. (*File Server*, *Memory Server*, etc)



Estrutura de um S.O.



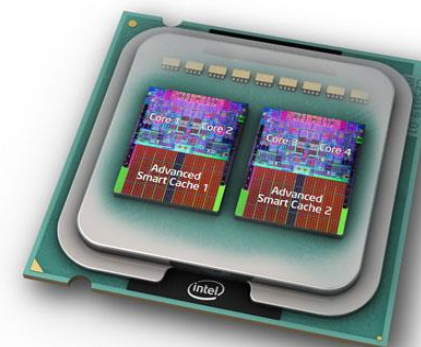
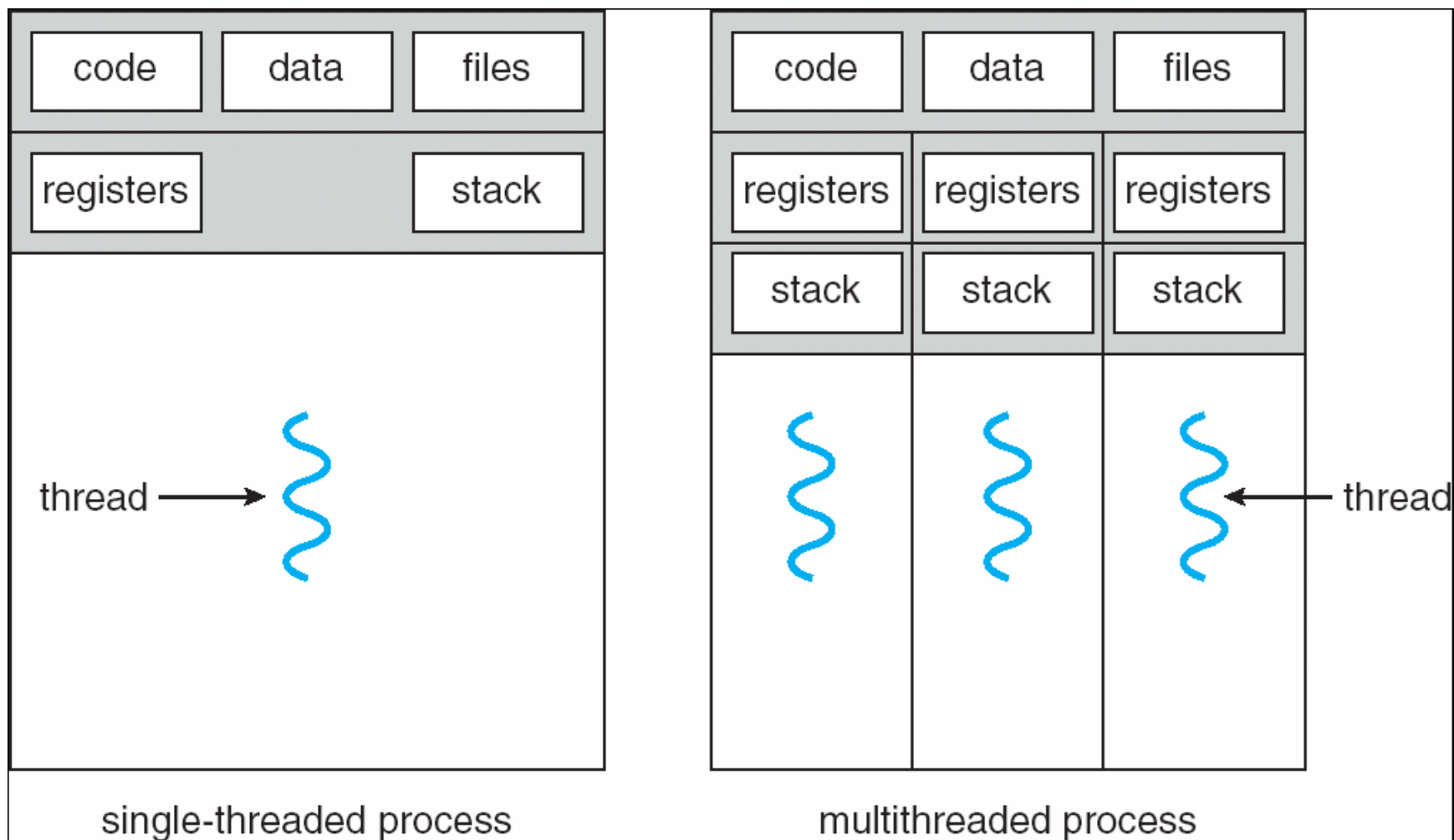
(+) Código do *kernel* é mais fácil de implementar e de elaborar

(-) Execução do código é menos eficiente devido às mudanças de contexto

- Conceitos introduzidos nos sistemas operativos modernos
 - Multi-threading
 - Object-Oriented Design
 - Symmetric MultiProcessing
 - Distributed Operating Systems

- Um processo é dividido num conjunto de *threads* que podem ser executados de forma concorrente
- Thread
 - Unidade de trabalho que pode ser escalonada
 - É executada sequencialmente e pode ser interrompida
 - Pertence a um processo
 - Processo é composto por uma ou mais *threads*
- Num sistema *monothread* (sistema normal) existe apenas uma thread de execução

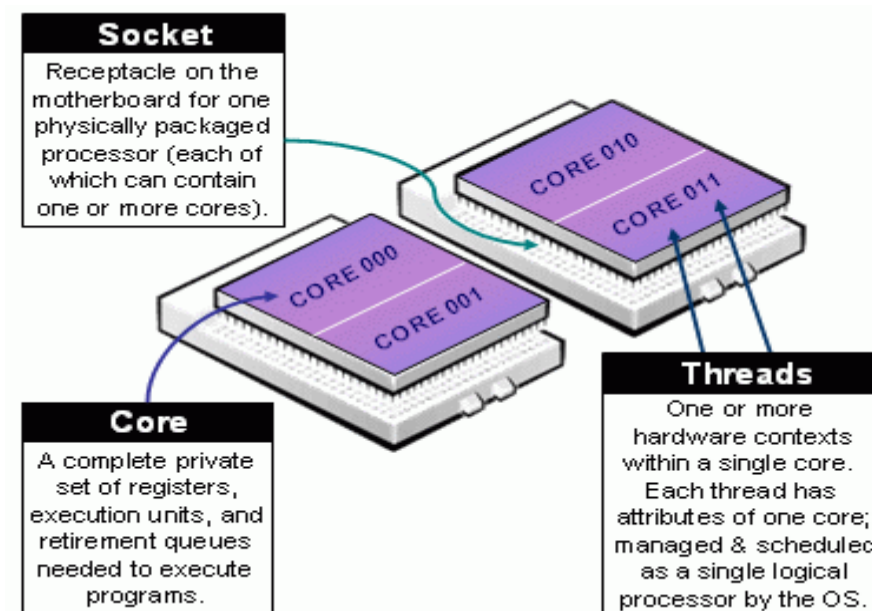
Multi-thread (exemplo)



- Aplicação dos paradigma dos objectos na concepção de sistemas operativos
- Permite adicionar extensões ao *kernel* de um sistema operativo na forma de módulos
- O programador pode “personalizar” um sistema operativo sem violar a integridade do sistema

Sistema SMP

- Vários processadores
 - Ou vários cores (“multi-core”)
- Cada processador partilha a mesma memória e os mesmos dispositivos de E/S
- Cada processador desempenha as mesmas funções



Symmetric multiprocessing (SMP)

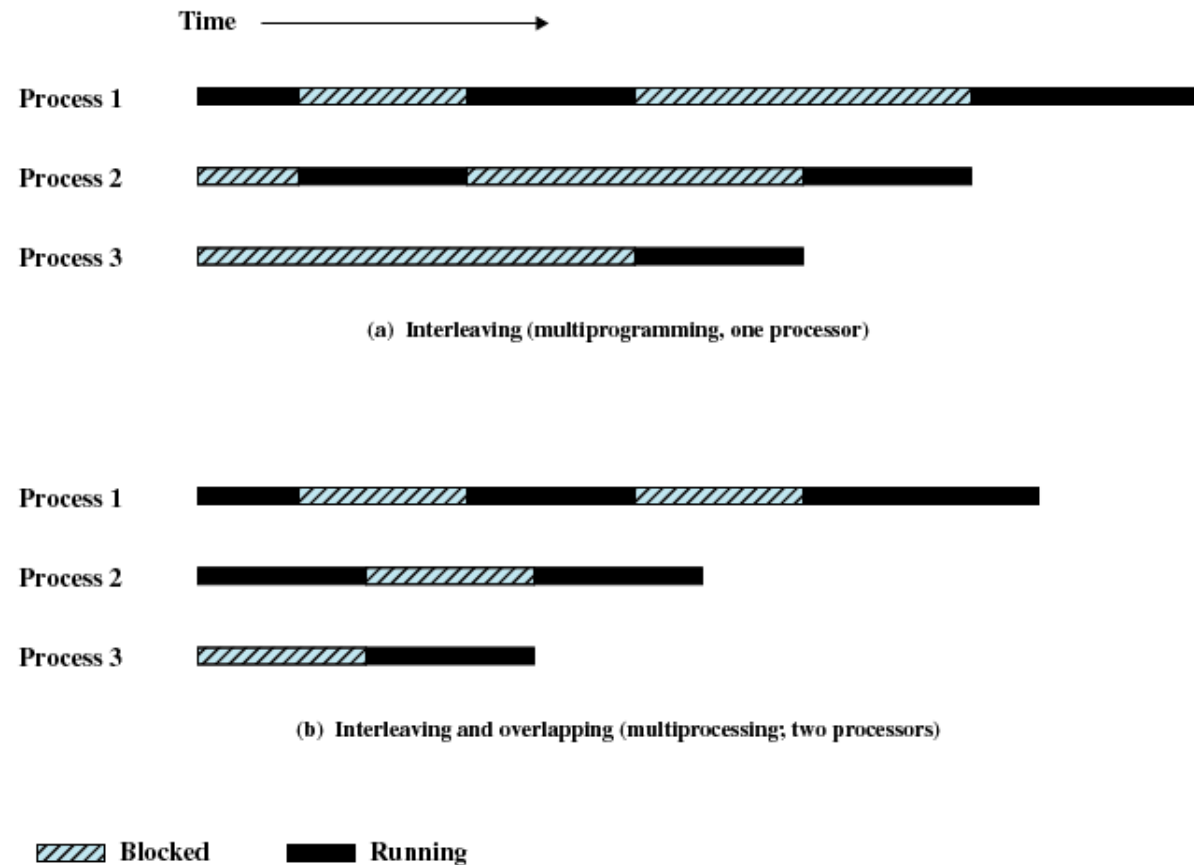
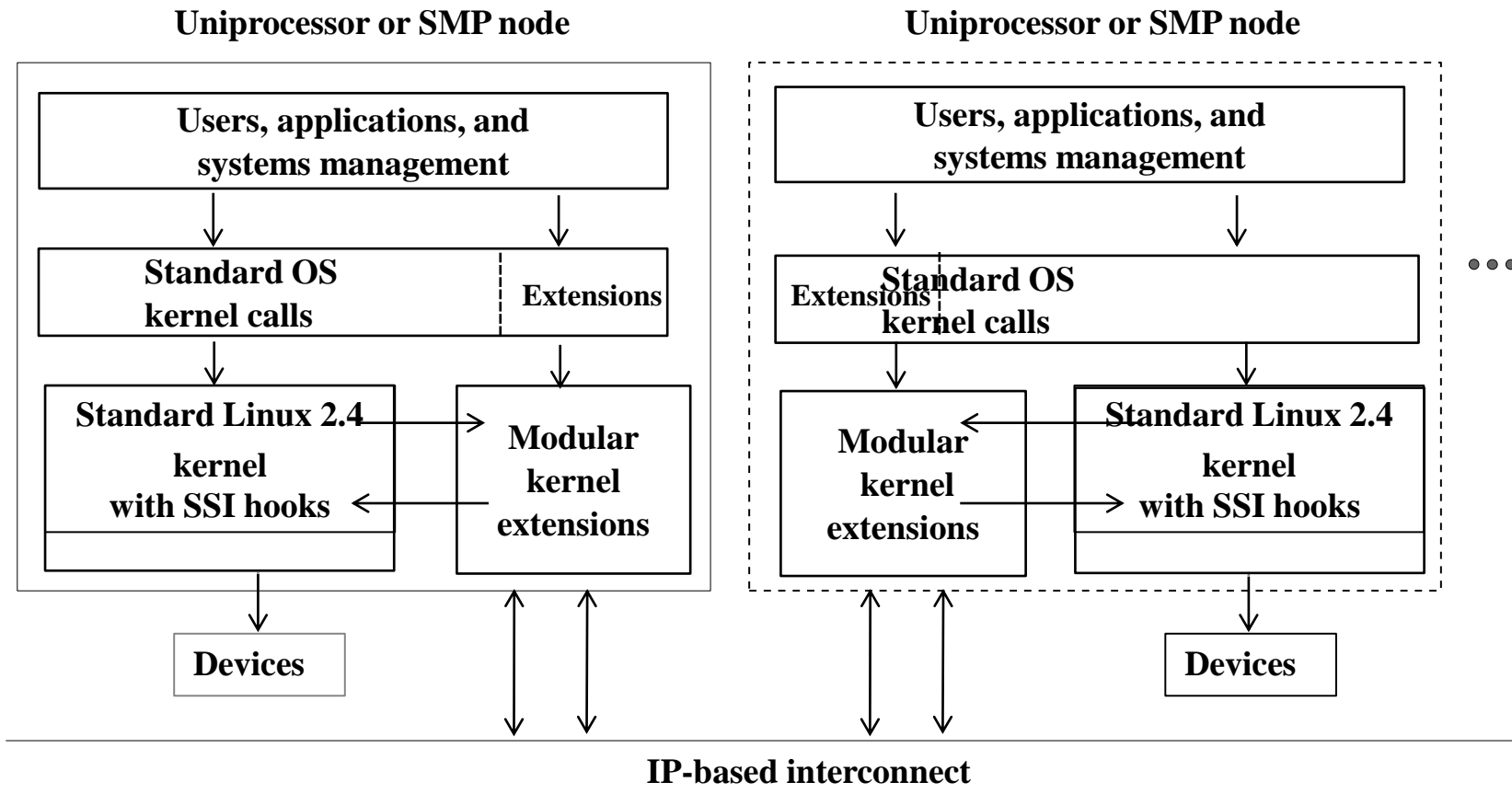


Figure 2.12 Multiprogramming and Multiprocessing

- Distribuição de todos os serviços providenciados pelo S.O.
- Visão unificada do sistema (*SSI – Single System Image*)
 - Espaço de endereçamento (\sum memória em cada nodo)
 - Número de processadores (\sum processadores em cada nodo)
- Exemplos:
 - OpenSSI
 - OpenMosix

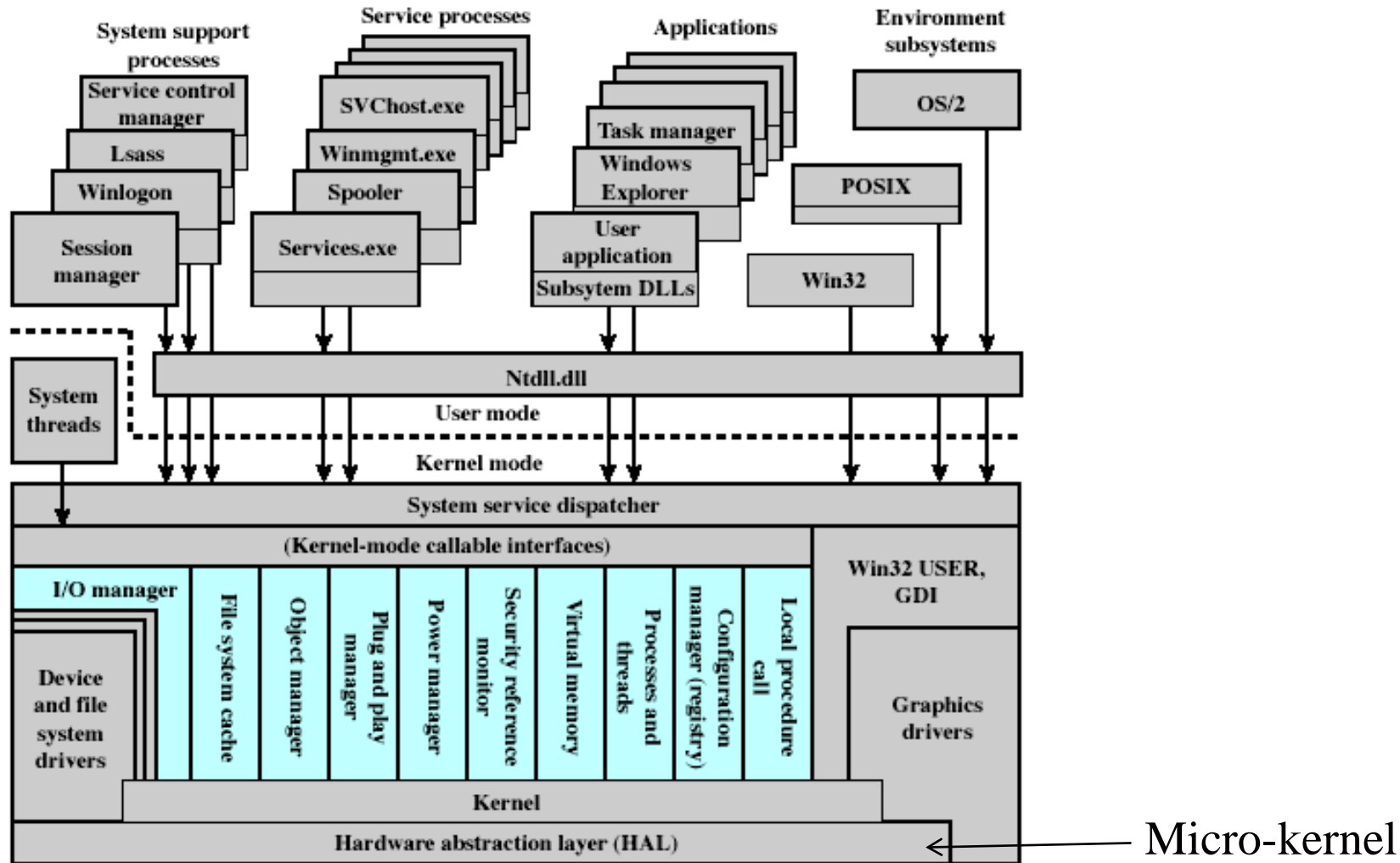
Distributed Operating Systems



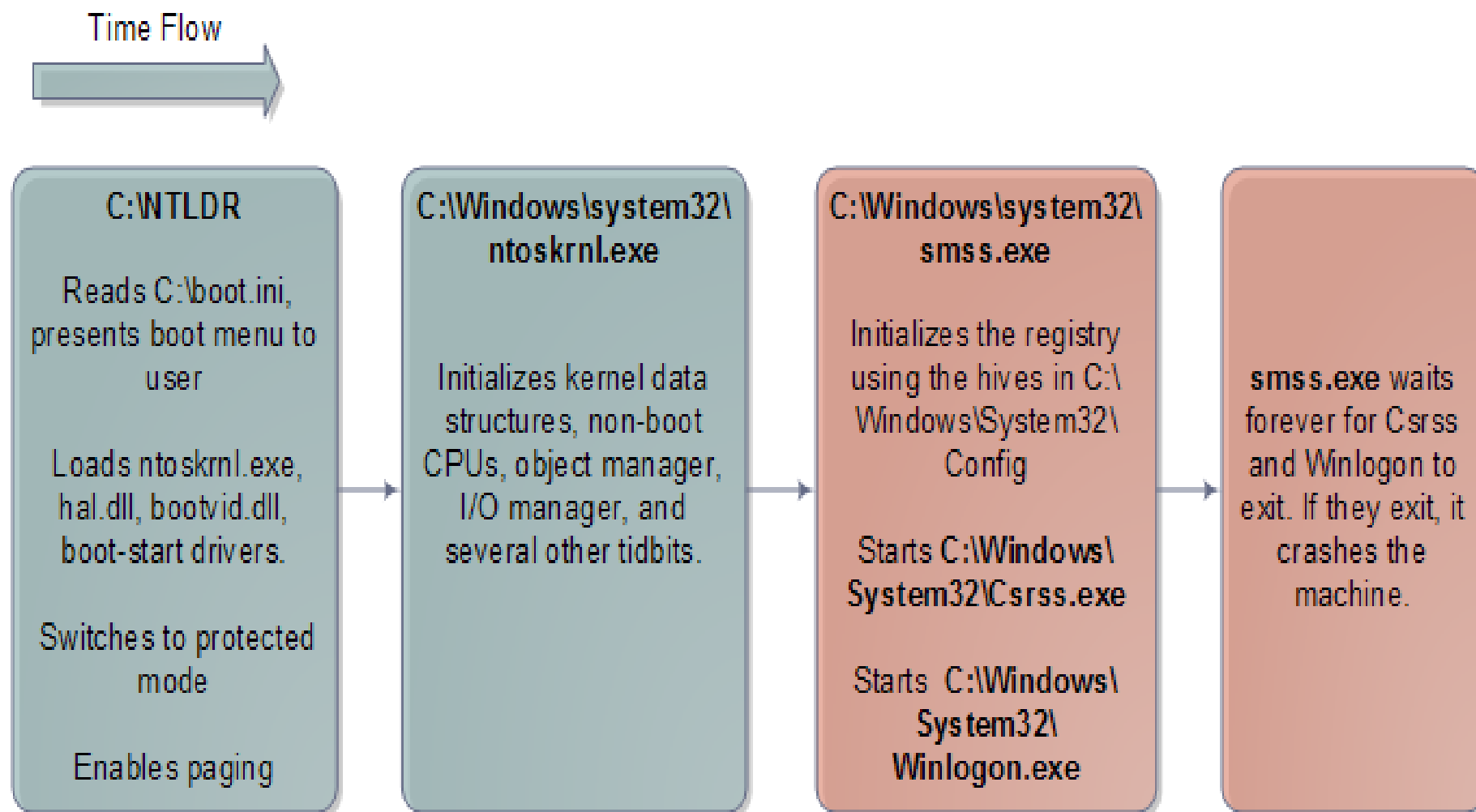
Windows 10, 11 / OS X Catalina

- *Single-user multitasking*
 - Um utilizador a executar múltiplos programas
- Estrutura micro-núcleo modificada
- Suporta aplicações escritas para outros sistemas operativos (DOS/Win9x)
- Serviços disponibilizados na forma de interfaces
- Serviços acedidos usando o modelo cliente/servidor
- Suporte para *threads* e SMP (ou multi-core)

Windows 10 e 2016 Server

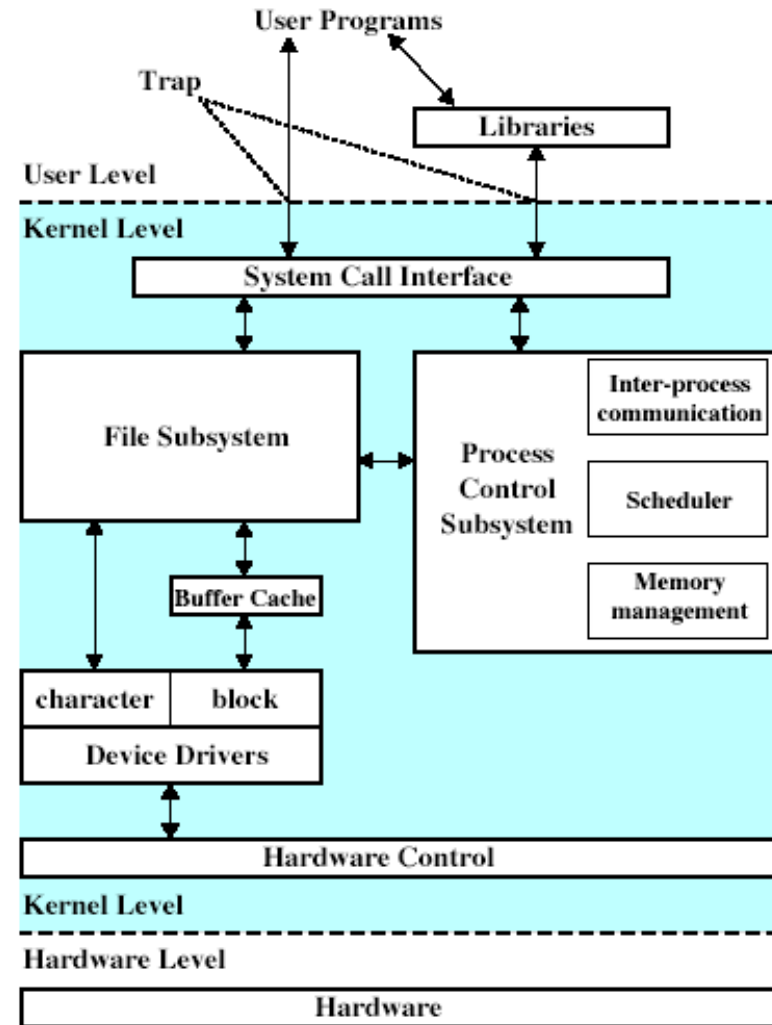
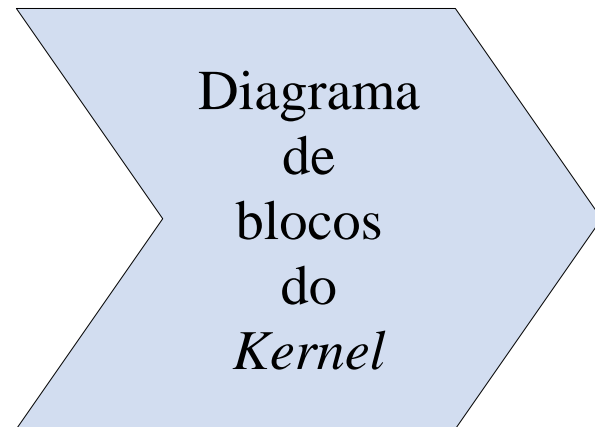


Arranque no Windows 10/11

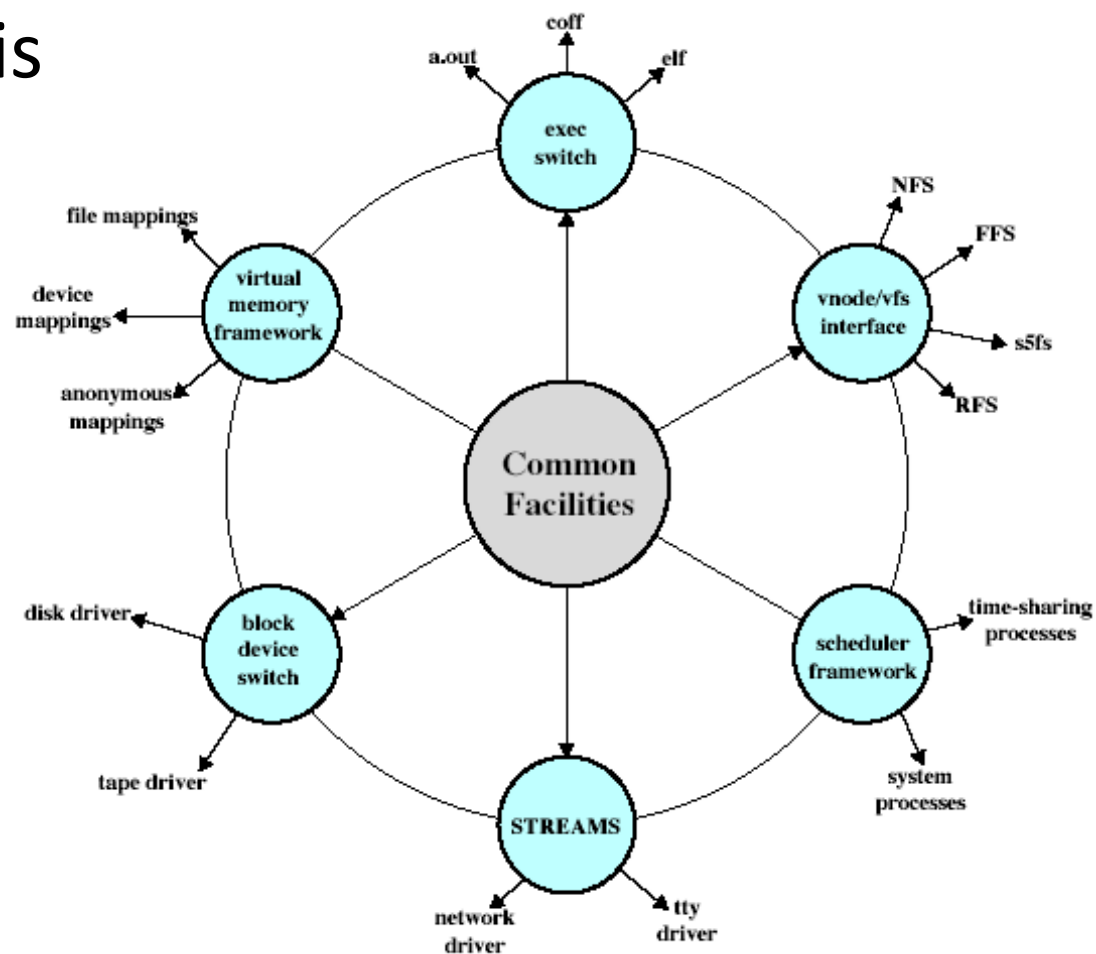


- O sistema operativo encapsula todo o hardware da máquina
 - S.O. é apelidado de System Kernel
- Serviços disponibilizados na forma de interfaces
- Normalmente segue uma estrutura monolítica
- Acompanhado por:
 - Shell
 - Conjunto de bibliotecas e utilitários

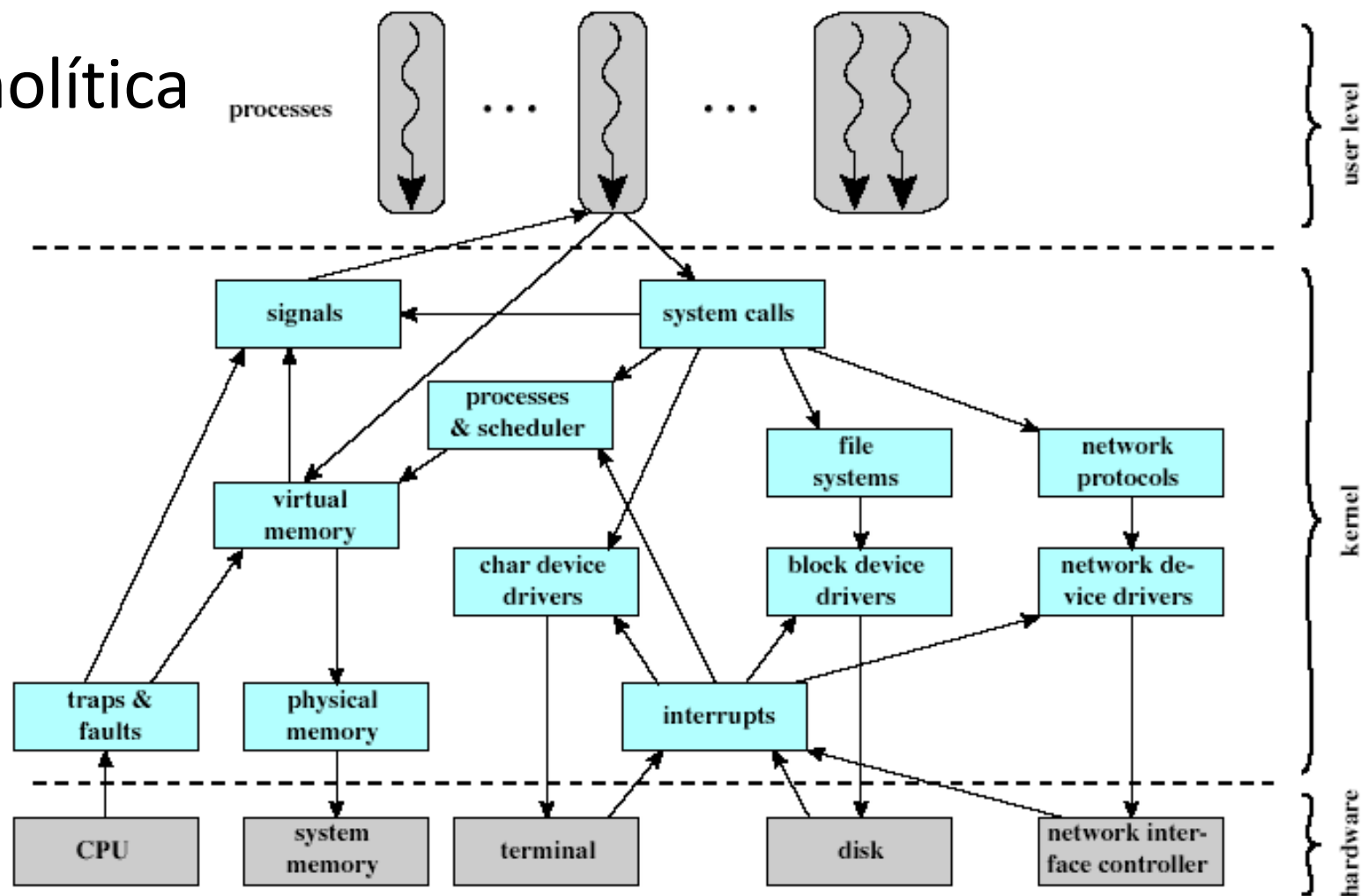
Estrutura tradicional



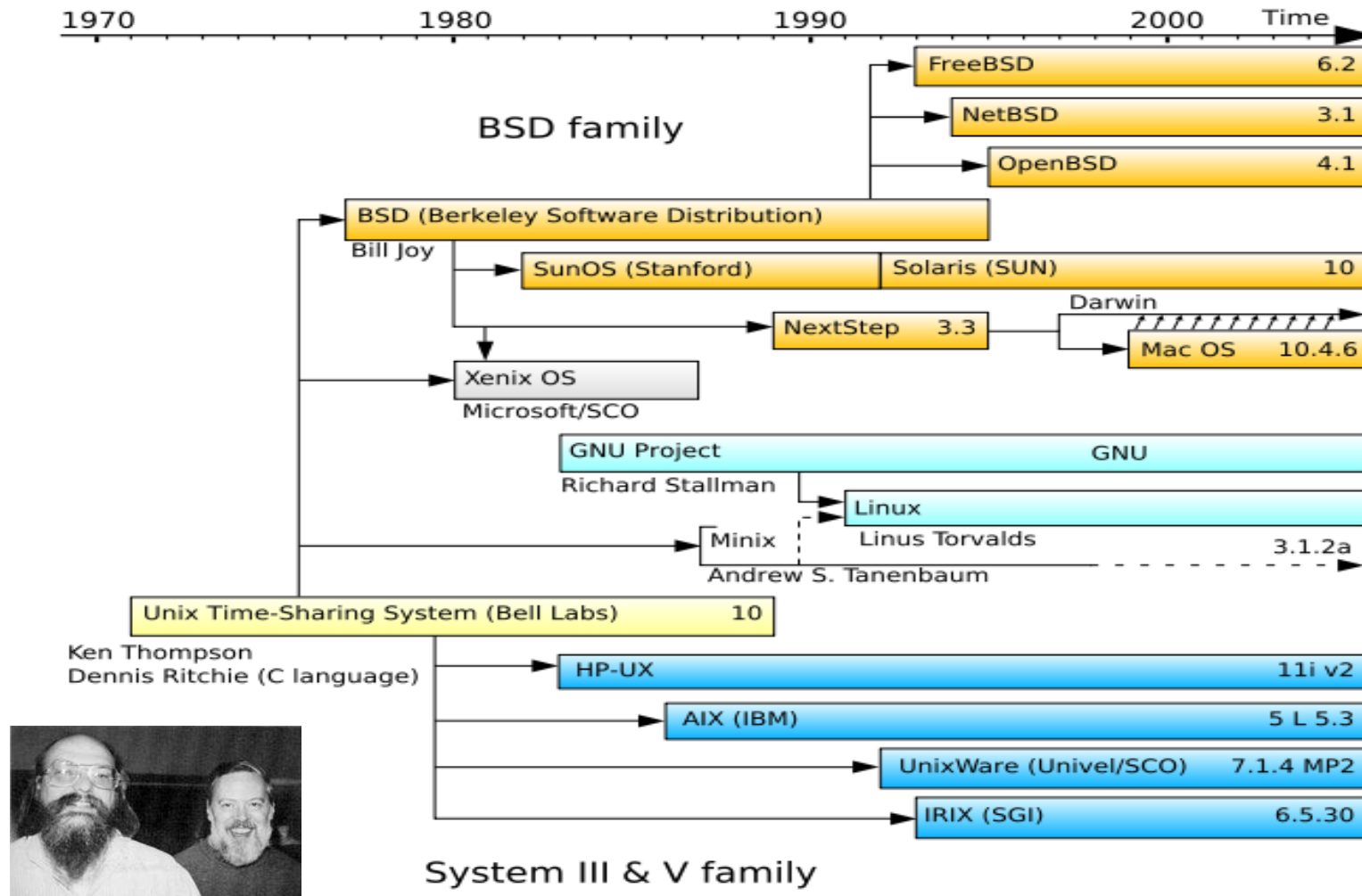
Estruturas actuais



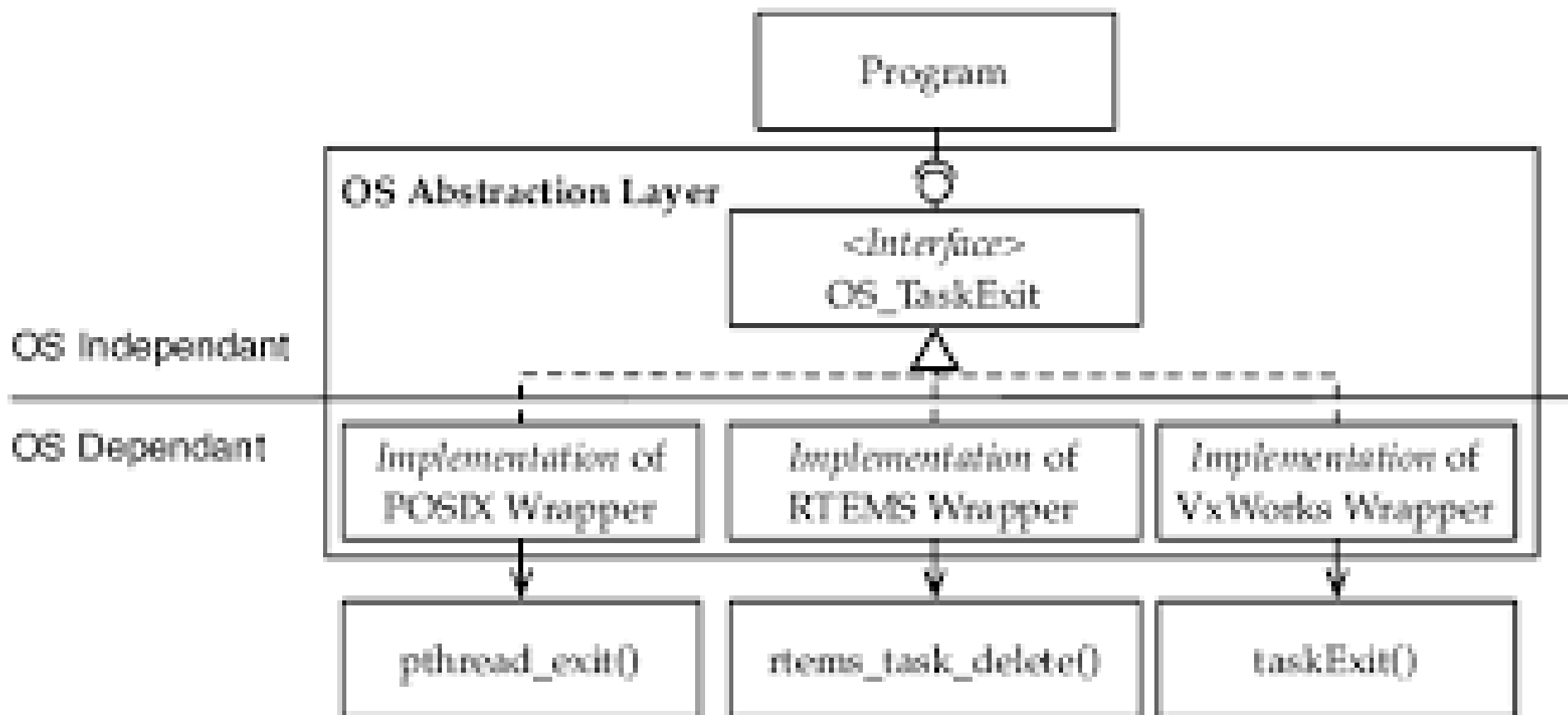
Estrutura monolítica



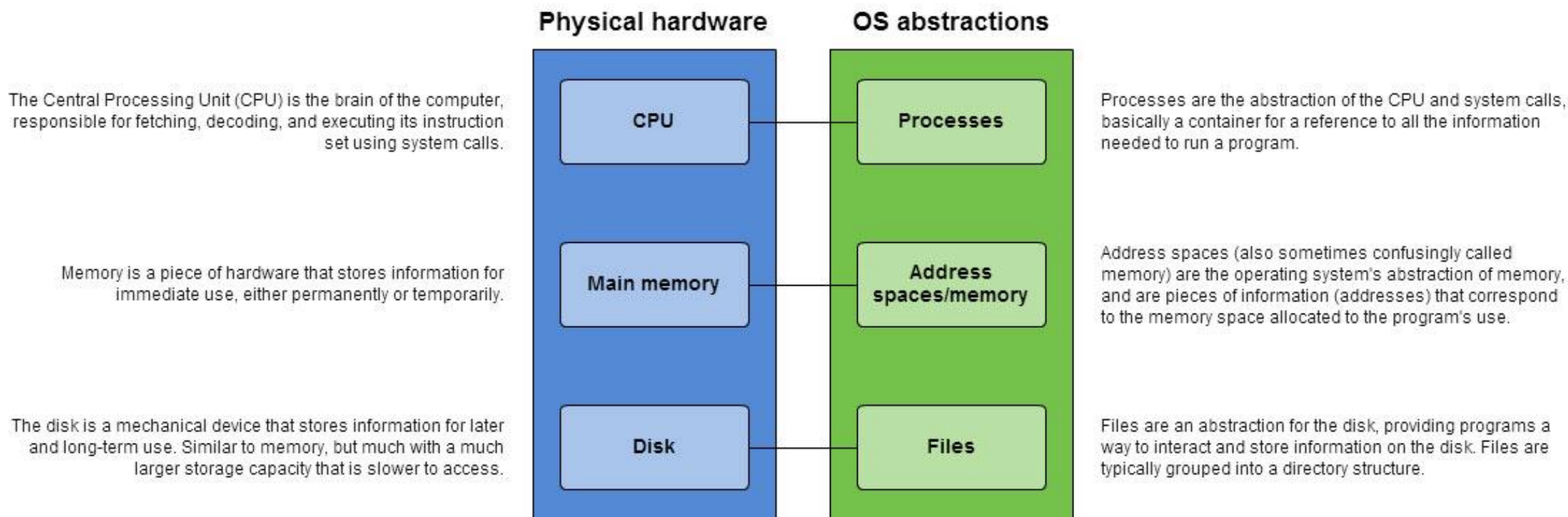
Unix vs Linux

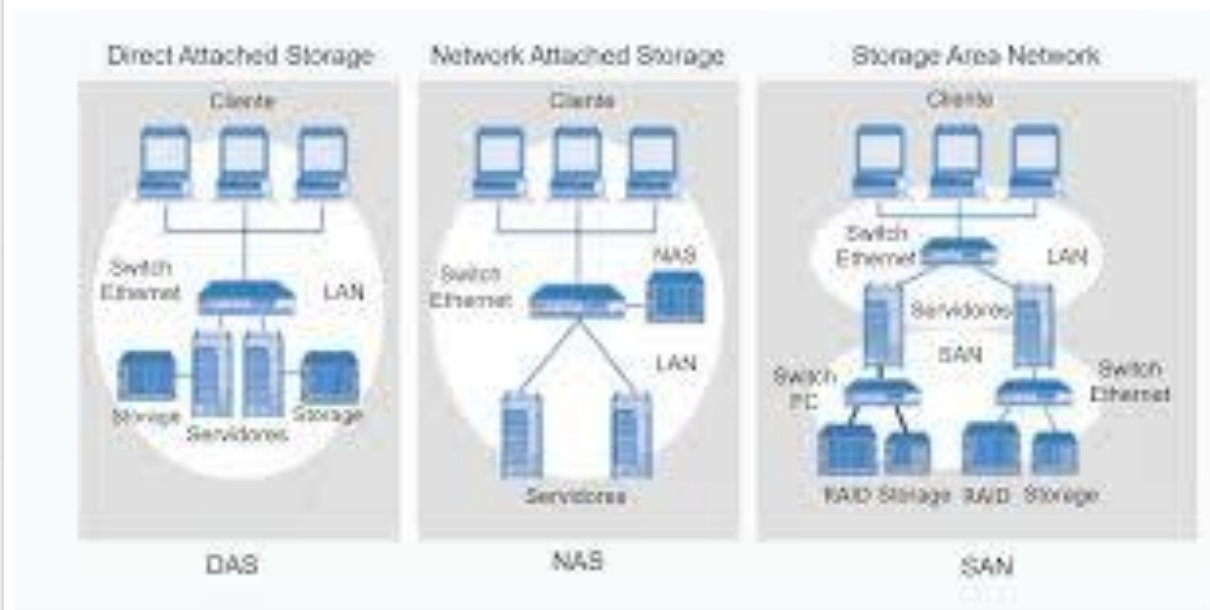
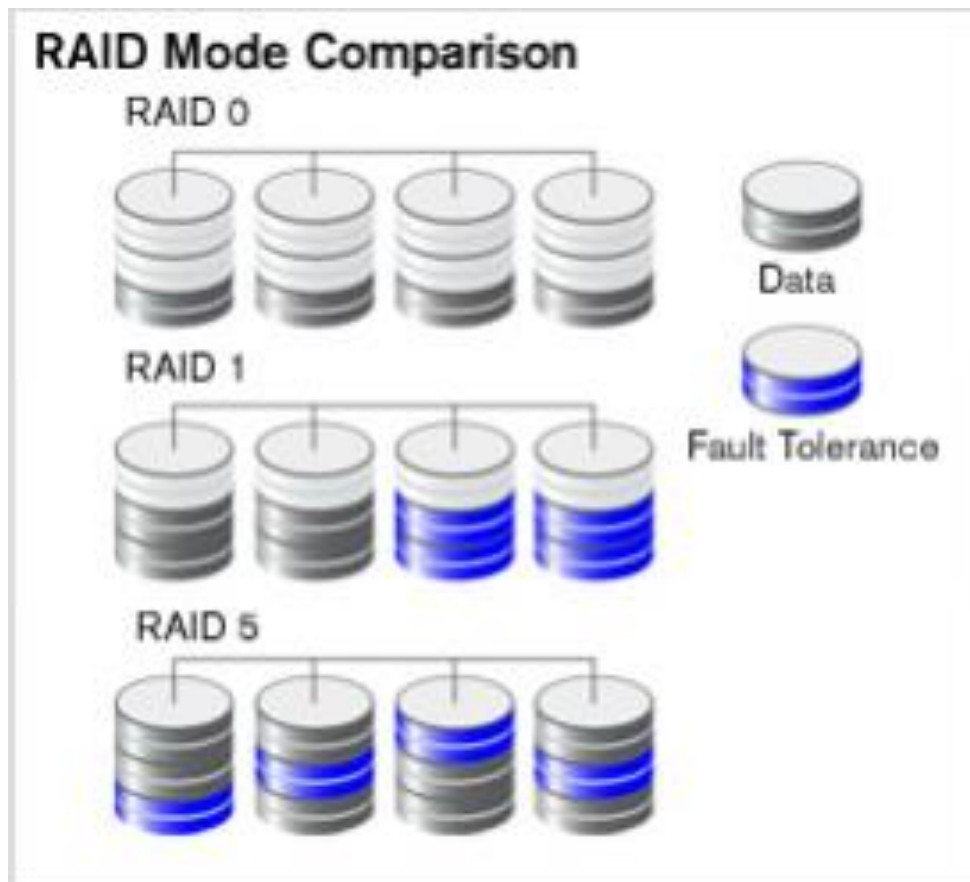


Abstrações suportadas por um S.O.



Abstrações suportadas por um S.O.





- Quais são os elementos chave de um Sistema Operativo?
- Qual é a estrutura de um S.O.?
- Distinguir processo de thread?
- Evoluções recentes dos sistemas operativos
- Que abstrações são suportadas por um S.O.?
- Que necessidades de suporte e de proteção de hardware existem?