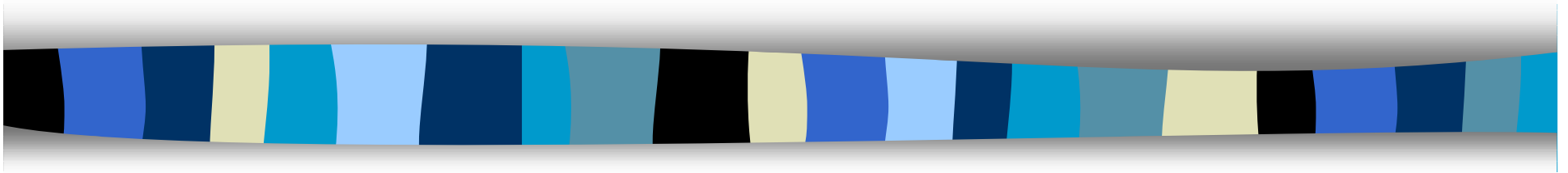


# Sistemas Operacionais



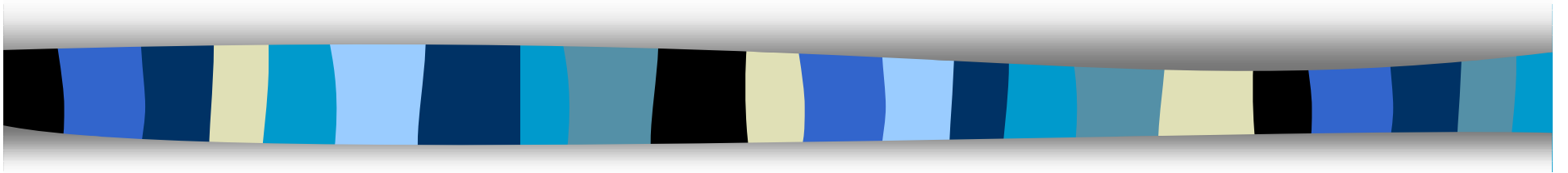
João Paulo Kitajima



# Sumário

- Conceitos Básicos
- Processadores e Processos
- Memória Primária e Dados
- Memória Secundária e Arquivos
- Periféricos e Entrada/Saída
  - Interfaces de rede e processos comunicantes
- Conclusões

# Conceitos Básicos

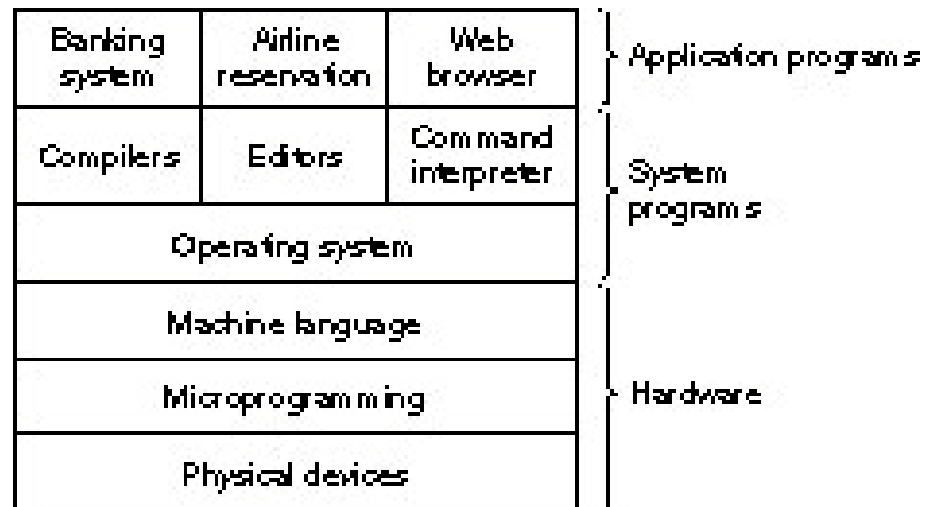




# Conceitos Básicos

- Um sistema computacional é formado por:
  - Um sistema **físico** (*hardware*): informalmente o “computador”
    - o monitor, a CPU, a RAM, o disco rígido, a placa de rede, a placa de fax/modem, o aparato de multimídia (alto-falantes, câmera), o teclado, etc...
  - Um sistema **lógico** (*software*): informalmente os “programas”
    - os aplicativos para usuários finais (ex. Microsoft Office, jogos), as linguagens de programação (ex. C, Fortran, Delphi), os sistemas de bancos de dados, as interfaces operacionais (ex. janelas, comandos textuais), os **sistemas operacionais** (ex. UNIX, Windows, MacOS), etc...

# Conceitos Básicos





# Conceitos Básicos

- Um sistema computacional também é um “sistema”, ou seja:
  - Recebe algum tipo de “estímulo” de seu ambiente (**entrada**, *input*)
  - Reage ao estímulo externo (realizando alguma **operação** e modificando o seu **estado**)
  - Produz um retorno para o ambiente em função do estímulo recebido (**saída**, *output*)



# Conceitos Básicos

## ■ Um sistema operacional:

- É um conjunto de programas (então é *software*) que executa imediatamente sobre o *hardware*
  - **Gerencia o hardware** (função “de cima para baixo”, associado ao fato de um dado computador executar um sistema operacional mais adequado)
  - Proporciona um modelo de **uso simples, virtual, do hardware complexo** (função “de baixo para cima”, associado ao fato de certos aplicativos executarem apenas sobre um dado sistema operacional)



# Conceitos Básicos

- O que o usuário/programador do computador faz?
  - Manda executar **instruções** que resolvem o seu problema (um editor de texto, um montador de genomas, um programa de alinhamento de seqüências, etc...)
  - Executa **chamadas de sistemas** (*system calls*) que são instruções de controle para o sistema operacional (independente da aplicação: iniciar um processo, emitir um sinal para finalizar um processo, criar um arquivo, criar um diretório/folder, proteger um arquivo, cronometrar um evento, etc...)

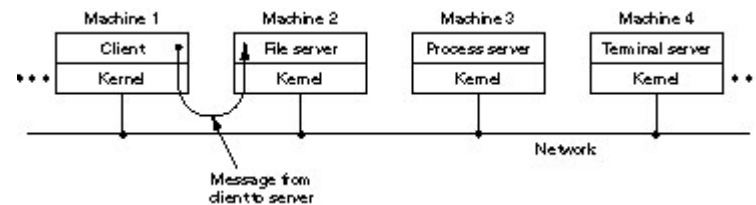
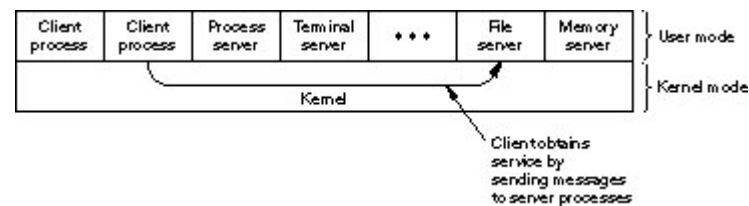
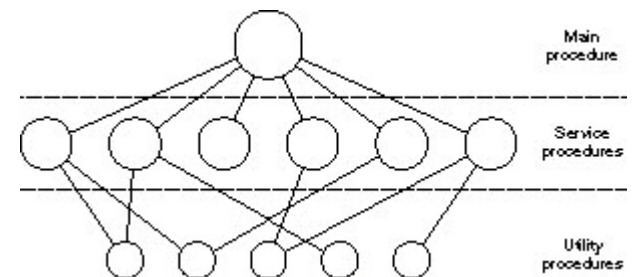
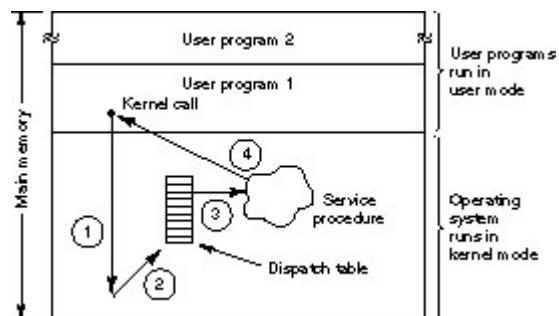




# Conceitos Básicos

- Estruturas possíveis (como qualquer outro software):
  - Sistemas **monolíticos**: um único programa, eventualmente procedural (organizado em diferentes rotinas)
  - Sistemas em **camadas** (o programa final do usuário chama rotinas de gerência de entrada/saída que chama rotinas da gerência de memória e disco que chama rotinas de alocação de processos e multiprogramação)
  - Sistemas **cliente-servidor** (duas camadas: [1] *kernel* e [2] clientes e servidores - processos do usuário solicitam serviços a servidores - o *kernel* cuida apenas da comunicação e da implementação de **mecanismos** básicos - a **política** é definida pelos servidores) - Windows, Linux, ...

# Conceitos Básicos





# Conceitos Básicos

- Do discurso anterior, percebemos o sistema:
  - Fala-se em **processos** (abstração da execução dinâmica de um **programa**)
  - Fala-se em **memória** (que guarda o estado de um processo, onde estado = dados+controle)
  - Fala-se em **entrada e saída**



# Conceitos Básicos

- Um Sistema Operacional permite um usuário:
  - Usar a **CPU** (processador) através dos **processos**
  - Usar a **RAM** (memória) através de mecanismos de gerência de **memória** (espaço de trabalho)
  - Usar **disco, monitor, teclado, impressora, mouse, etc...** através de gerenciadores de entrada e saída (= **drivers**)
    - A “**rede**” pode ser vista como um dispositivo de E/S: você escreve na rede (enviando mensagens) e você lê da rede (recebendo mensagens)
  - Um SO deve prover mecanismos básicos e políticas para tarefas fundamentais (como uso do processador)



# Conceitos Básicos

## ■ Windows versus Linux:

- Windows: atende bem uma das funções de um SO - prover um computador fácil de usar. Porém, sistema fechado
- Linux: atende bem a outra função de um SO - gerenciar recursos físicos (CPU, memória, etc...)

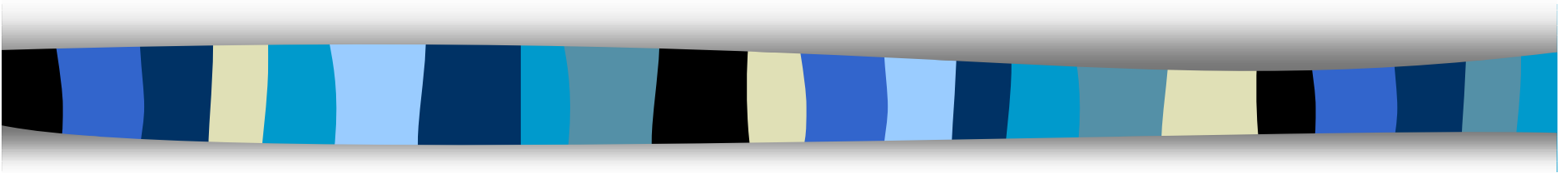
Os conhecidos erros de Windows geralmente não acontecem em Linux. Por outro lado, Linux tem dificuldade de prover facilidades para desenvolvimento de aplicativos básicos eficazes (como editores de texto). Windows está amarrado a um tipo de processador. SO's do tipo Unix são mais genéricos em termos de processador de base.



# Conceitos Básicos

- Um bioinformata ao instalar e usar um SO:
  - Deve saber o máximo possível sobre **especificações** do hardware (fabricante e modelo são essenciais)
  - Deve conhecer funcionalidades básicas do SO para poder avaliar o seu uso em aplicações de bioinformática: isto significa que deve saber pelo menos **quais SO's suportam quais softwares de bioinformática**
  - Deve conhecer as **abstrações que o SO suporta** (ex. processos) e principalmente como obter e avaliar informações sobre estas abstrações

# Processos e Processadores





# Processos e Processadores

- **Algoritmo**: “receita” em passos para resolver um problema (por exemplo, computacional)
- **Programa**: conjunto de instruções escritas em uma linguagem de programação compreensível por um sistema computacional. Implementa um algoritmo
- **Processo**: abstração de um programa em execução
  - instrução em execução
  - valor do contador de programa (endereço da próxima instrução a ser executada)
  - conteúdo de registradores e variáveis (espaço de trabalho)





# Processos e Processadores

## ■ Processos:

- Têm entradas
- Gera saídas
- Têm um estado
- Conversa direta ou indiretamente com o sistema operacional através de chamadas de sistemas (*system calls*)



# Processos e Processadores

## ■ Multiprogramação

- Em um sistema computacional, existem vários programas executando, ou seja, existem vários processos ativos (e.g., um editor de texto, um navegador da Web e o próprio sistema operacional)
- Se o seu computador tem apenas um Pentium™, como ele pode suportar vários processos executando ao mesmo tempo? Ele faz um **chaveamento** entre os mesmos: isto é multiprogramação (impressão de várias CPUs virtuais)



# Processos e Processadores

## ■ Paralelismo e Multiprocessadores

- E se o computador tem mais de uma CPU, ou seja, é um sistema **multiprocessado**? Podemos então ter mais de um processo ativo ao mesmo tempo em processadores reais - isto é **paralelismo**
- Qual a vantagem? Mais processadores, maior velocidade de processamento
- Qual o cuidado? Se houver apenas uma memória acessível pelos vários processadores, temos conflito

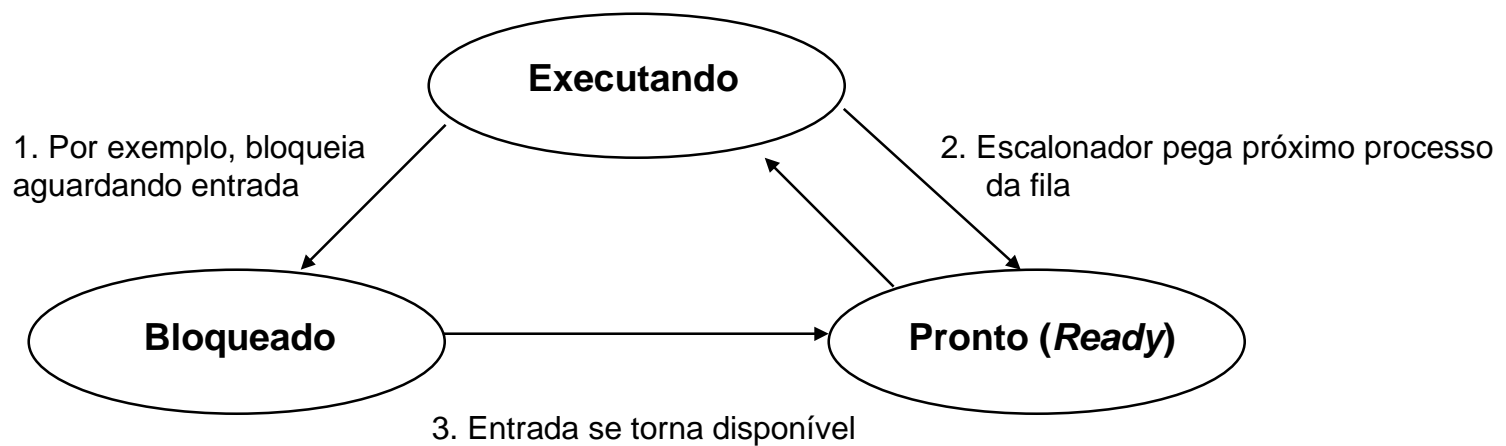


# Processos e Processadores

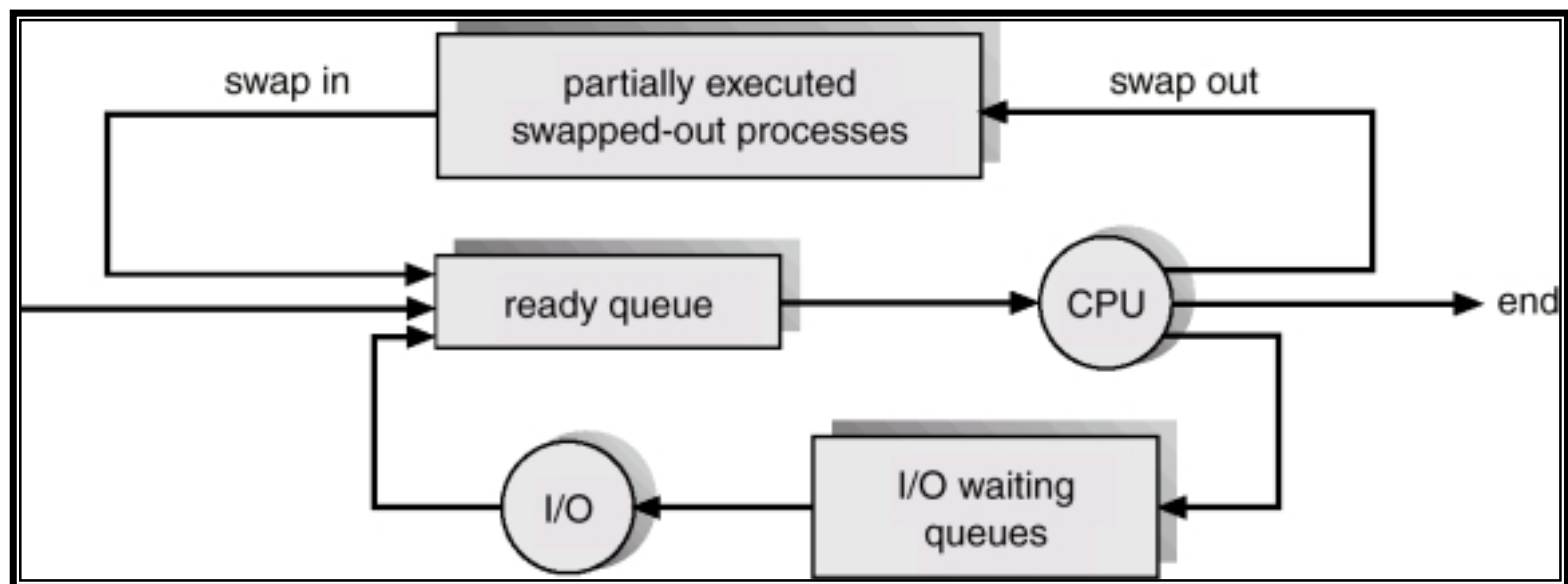
- Processos podem criar outros
- Um sistema operacional implementa e gerencia uma árvore de processos
- *init* é em geral o primeiro processo
- Existem chamadas de sistemas para criar processos (*fork*)
- **Sinais** podem ser enviados a um processo para terminá-lo antes do tempo

# Processos e Processadores

- Processos têm estados: **executando**, **pronto** para executar, **bloqueado**

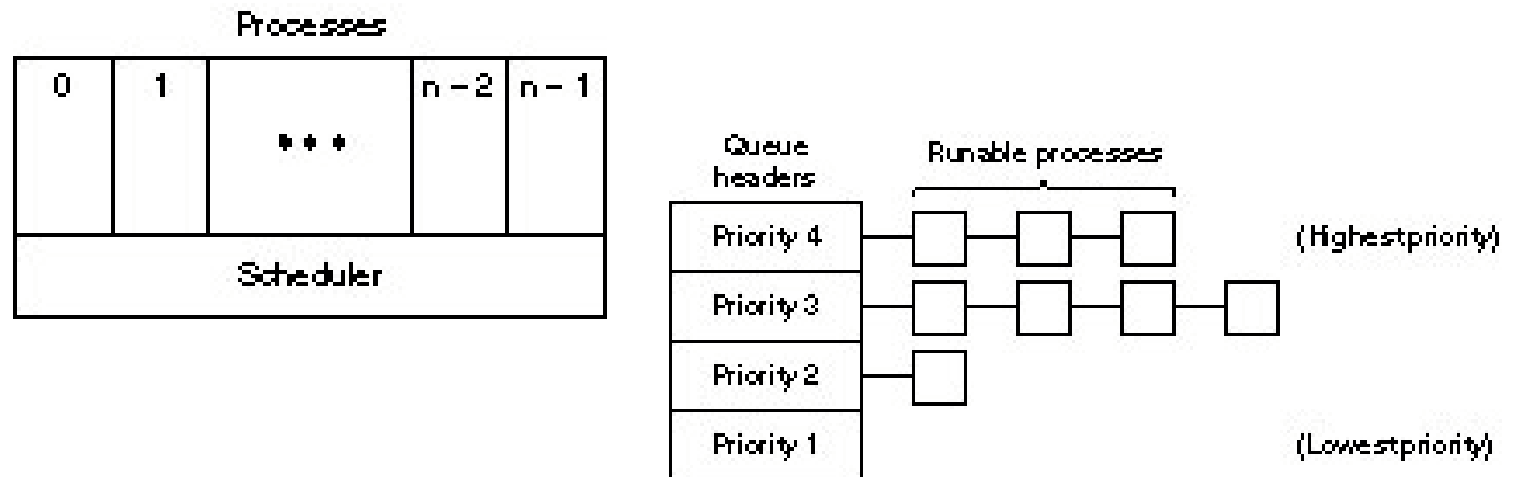


# Processos e Processadores



# Processos e Processadores

- Problema básico: quem deve executar, visto que há vários processos prontos para executar?  
Resposta: o escalonador (*scheduler*)





# Processos e Processadores

## ■ Tabela de Processos

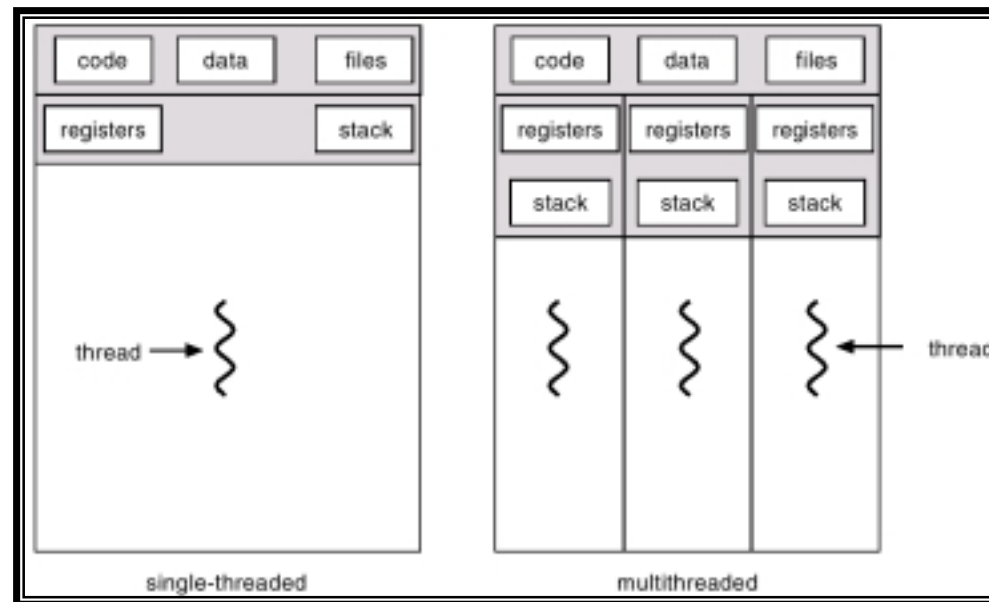
- Processos são controlados através de uma tabela que contém informações dinâmicas e estáticas dos processos. O comando “ps” do Unix nada mais faz do que listar esta tabela (veja fim desta seção)



# Processos e Processadores

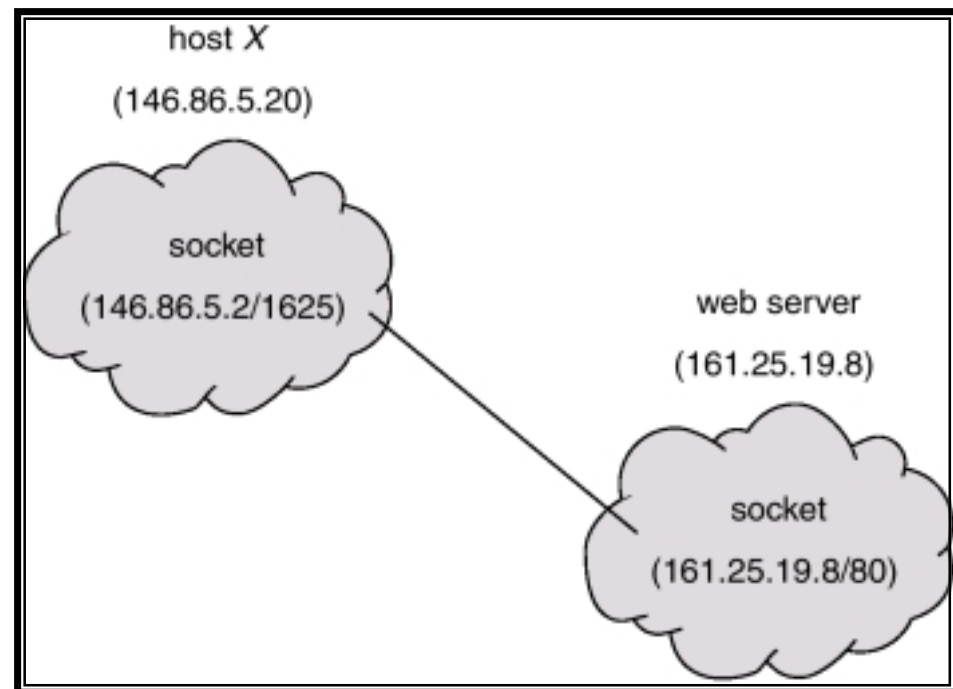
## ■ Processos Leves (*Threads*)

- exemplo: blast de seqüências via linha de comando
- `blastall -i query.ptn -o saida -p blastp -d nr -a 4`



# Processos e Processadores

- Processos se comunicam





# Processos e Processadores

- Vendo processos em UNIX
  - Comando ps (comando padrão do Unix)

jpk@onsona% ps -elf | more

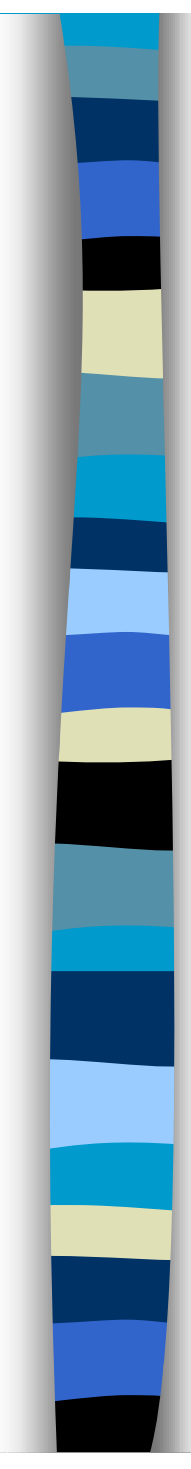
F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	CMD
19	T	root	0	0	0	0	SY	10417f10	0		Mar 04	?	0:28	sched
8	S	root	1	0	0	41	20	629a5608	88	629a5800	Mar 04	?	0:02	/etc/init -
19	S	root	2	0	0	0	SY	629a4f48	0	104396f8	Mar 04	?	0:00	pageout
19	S	root	3	0	1	0	SY	629a4888	0	1043c934	Mar 04	?	208:18	fsflush
8	S	root	319	1	0	40	8	62bb9610	241	62bb9808	Mar 04	?	0:00	/usr/lib/inet/xntpd
8	S	root	102	1	0	40	20	629a26c8	827	ef4e1e7c	Mar 04	?	1:44	/usr/local/net/sbin/named -c /etc/n
8	S	root	522	485	0	59	20	630b0030	769	630b00a0	Mar 04	?	0:00	/usr/dt/bin/dtlogin -daemon
8	S	root	158	1	0	41	20	62bb7b10	268	62a6ba3e	Mar 04	?	0:01	/usr/sbin/inetd -s
8	S	root	114	1	0	41	20	62bb81d0	253	62a6bd0e	Mar 04	?	0:00	/usr/sbin/keyserv
8	S	daemon	164	1	0	41	20	62bb66d0	288	ef30de7c	Mar 04	?	0:00	/usr/lib/nfs/statd
8	S	root	125	1	0	41	20	62c70f58	246	62a6bc96	Mar 04	?	0:49	/usr/lib/netsvc/yp/ypserv -d
8	S	root	112	1	0	41	20	62bb8f50	264	62a6bdd6	Mar 04	?	0:08	/usr/sbin/rpcbind
8	S	root	127	125	0	41	20	629a2d88	198	62a6bc6e	Mar 04	?	0:01	rpc.nisd_resolv -F -C 8 -p 10737418
8	S	root	135	1	0	55	20	62c6f458	226	62a6bbce	Mar 04	?	0:00	/usr/lib/netsvc/yp/ypxfrd
8	S	root	132	1	0	41	20	62c6fb18	237	62a6bba6	Mar 04	?	0:00	/usr/lib/netsvc/yp/ypbind
8	S	root	321	1	0	41	20	62f3a6e8	276	62a6b3d6	Mar 04	?	0:03	/usr/lib/sendmail -C /etc/mail/send
8	S	root	142	1	0	95	20	62c6e6d8	205	62a6bab6	Mar 04	?	0:00	/usr/lib/netsvc/yp/rpc.yppupdated
8	S	root	140	1	0	85	20	629a41c8	220	62a6bb56	Mar 04	?	0:00	/usr/lib/netsvc/yp/rpc.yppasswdd -m
8	S	setubal	521	485	0	40	20	62bb7450	4020	62a6b4ee	Mar 04	?	11:00	/usr/openwin/bin/Xsun :0 -nobanner
8	S	root	175	1	0	41	20	629a3448	223	62a6b9ee	Mar 04	?	0:00	/usr/lib/nfs/lockd
8	S	root	197	1	0	41	20	629a2008	353	629a26b0	Mar 04	?	2:25	/usr/lib/autofs/automountd
8	S	root	201	1	0	41	20	62c6e018	597	eef05e7c	Mar 04	?	0:04	/usr/sbin/syslogd
8	S	root	300	1	0	41	20	62f3c1e8	297	62a6af26	Mar 04	?	0:01	/usr/sbin/vold
8	S	root	240	1	0	47	20	62daaf60	308	62dab158	Mar 04	?	0:05	/usr/sbin/nsd
8	S	root	234	1	0	51	20	62c70898	204	6023deb8	Mar 04	?	0:00	/usr/sbin/cron
8	S	daemon	348	347	0	40	20	62bb8890	812	62a6b106	Mar 04	?	0:00	/usr/lib/ab2/dweb/sunos5/bin/dwhttp
8	S	root	492	1	0	41	20	631aef78	236	62a6a7f6	Mar 04	?	0:00	/usr/lib/snmp/snmpdx -y -c /etc/snm
8	S	root	809	389	0	66	20	633b5488	390	62a6a27e	Mar 04	?	16:58	/usr/local/net/sbin/sshd
8	S	root	349	323	0	71	30	62daa1e0	508	62a6b336	Mar 04	?	0:06	/usr/sbin/nsrmmdbd
8	S	root	255	1	0	41	20	62bb6010	336	62a6b926	Mar 04	?	0:00	/usr/lib/lpsched
8	S	root	439	323	0	40	5	62bb6d90	463	62a6abb6	Mar 04	?	0:35	/usr/sbin/nsrmmmd -n 2
8	S	root	291	1	0	41	20	62f3cf68	116	62a6b386	Mar 04	?	0:00	/usr/lib/utmpd
8	S	root	323	1	0	41	20	62da8da0	577	62a6ac2e	Mar 04	?	0:17	/usr/sbin/nsrd
8	S	root	280	1	0	49	20	62da86e0	152	62da88d8	Mar 04	?	0:00	/usr/lib/power/powerd
8	S	root	356	1	0	99	20	62c6ed98	122	62c6ee08	Mar 04	?	0:00	/bin/sh /usr/local/bin/safe_mysqlld
8	S	root	310	1	0	41	20	62f3c8a8	304	62a6b30e	Mar 04	?	0:01	/usr/sbin/nsrexecd
8	S	root	314	310	0	41	20	62f3b468	326	62a6b156	Mar 04	?	0:01	/usr/sbin/nsrexecd



# Processos e Processadores

- Vendo processos em Unix
  - Comando top (programa desenvolvido por William Lefebvre). Consultar:

<http://www.groupsys.com/topinfo/>



```

last pid: 2661; load averages: 0.00, 0.01, 0.01
17:54:55
105 processes: 103 sleeping, 1 zombie, 1 on cpu
CPU states: % idle, % user, % kernel, % iowait, %
swap
Memory: 4096M real, 1376M free, 81M swap in use, 939M swap free

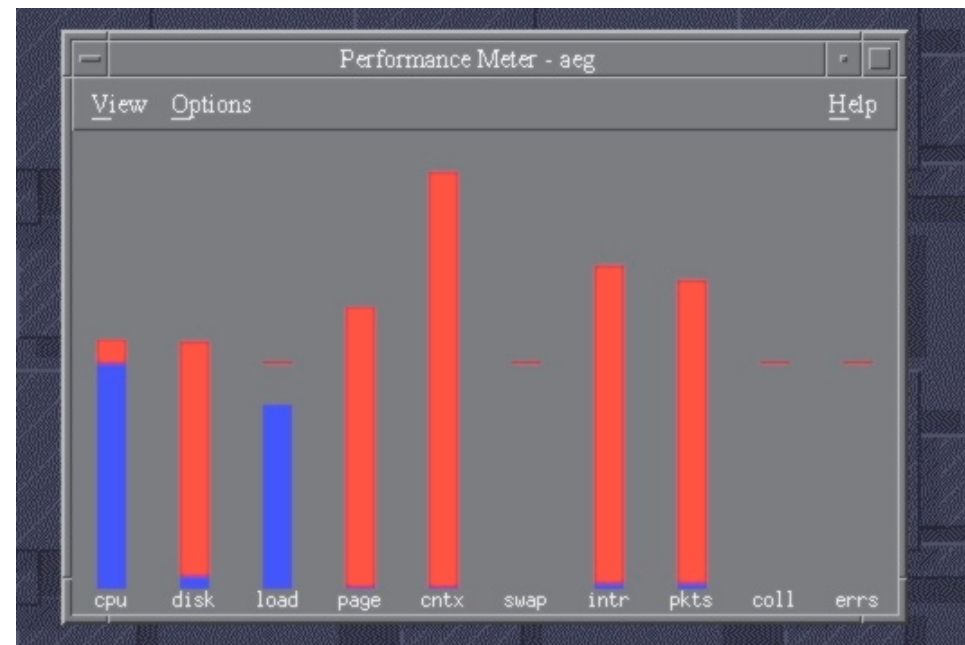
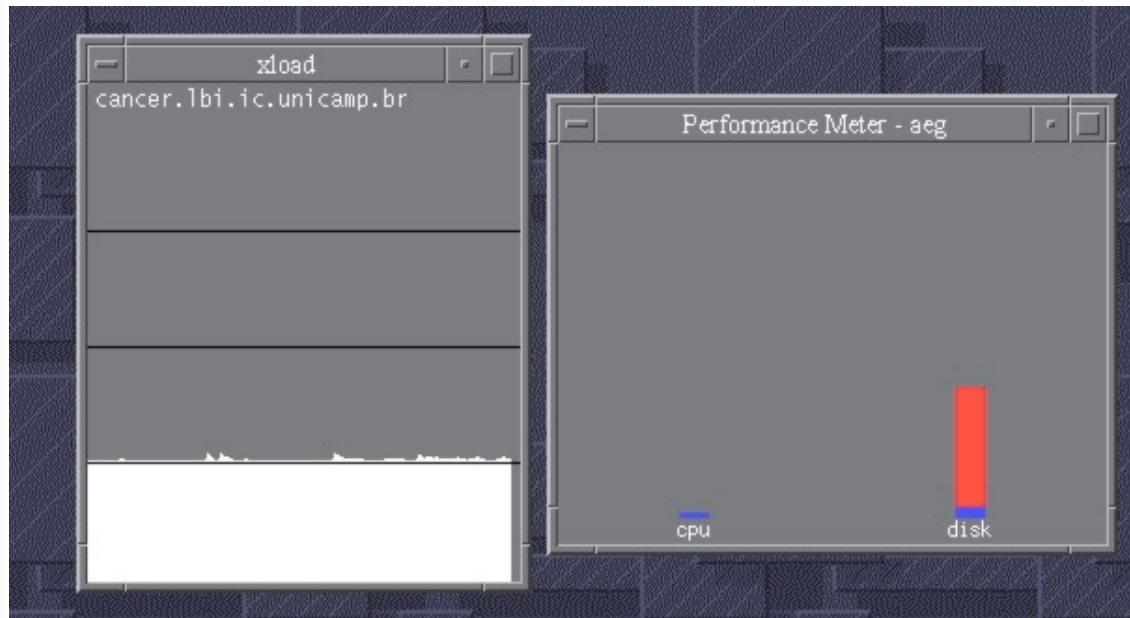
  PID USERNAME THR PRI NICE  SIZE  RES STATE  TIME  CPU COMMAND
2661 jpk      1   0   0 1608K 1472K cpul   0:00 0.38% top
 399 root      1  58   0 2272K 1392K sleep 10:41 0.04% _upsd
 400 root      2  59 -15 4464K 2880K sleep 28:30 0.00% nsrmmmd
 809 root      1  33   0 3120K 2248K sleep 16:57 0.00% sshd2
 521 root      1  59   0   31M   28M sleep 11:00 0.00% Xsun
2911 setubal   1  48   0   40M   30M sleep  7:17 0.00% netscape
 398 root      1  28  10 4424K 2840K sleep  6:44 0.00% nsrindexd
 197 root      6  58   0 2824K 2024K sleep  2:25 0.00% automountd
 102 root      9  59   0 6616K 4424K sleep  1:44 0.00% named
1029 setubal   1  59   0   17M   11M sleep  1:27 0.00% emacs
 125 root      1  58   0 1968K 1392K sleep  0:49 0.00% ypserv
 439 root      2  59 -15 3704K 2672K sleep  0:34 0.00% nsrmmmd
17854 katsumi    1  46   4   20M   16M sleep  0:27 0.00% netscape
 323 root      1  58   0 4616K 3392K sleep  0:17 0.00% nsrd
5995 katsumi    4  47   4   19M   14M sleep  0:14 0.00% acroread

```



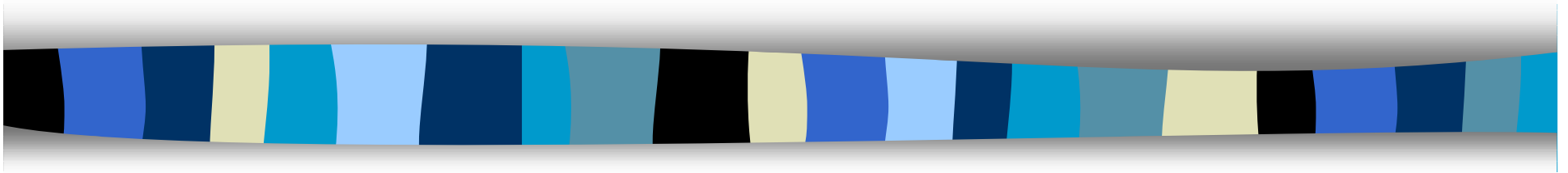
# Processos e Processadores

- Vendo processos em Unix
  - Comando gráfico (ex. xload em Solaris/Sun)





# Memória Primária e Dados





# Memória Primária e Dados

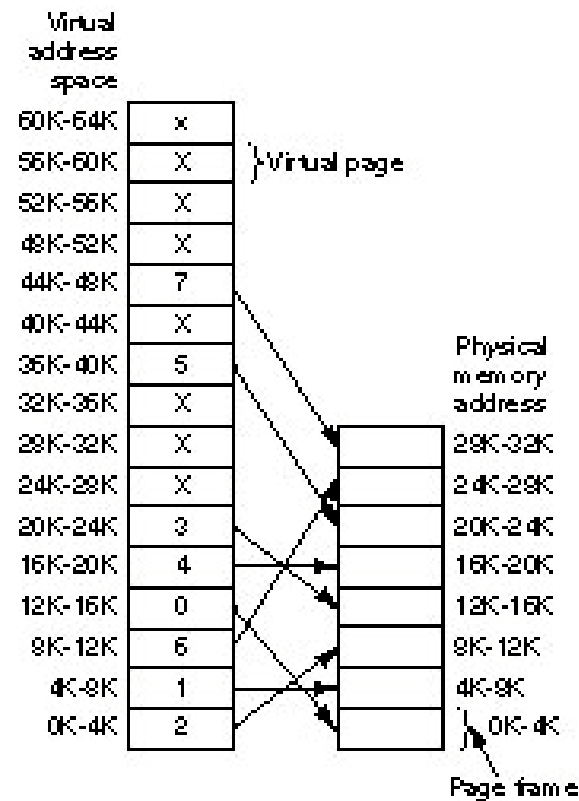
- Processos são compostos em parte por:
  - Instruções e
  - Dados
- Os dados compõem uma área de trabalho acessível em um local chamado de *memória*
- Fisicamente, esta memória pode ser uma RAM (*random access memory* - rápido acesso, acesso direto não sequencial, mas cara, limite: gigabytes) ou um disco (acesso lento, acesso semidireto, mas barato, limite: terabytes)



# Memória Primária e Dados

- E por que não colocar tudo na RAM?
  - Se o sistema comporta muitos processos, a soma da área de trabalho de todos eles é menor do que a capacidade da memória RAM (também chamada de primária)
- O que se faz?
  - Parte da área de trabalho de cada processo fica na RAM e parte reside em disco (*swap*). Chamamos este uso conjunto RAM/disco de **memória virtual**. **O ideal é que tudo fique na RAM.**

# Memória Primária e Dados

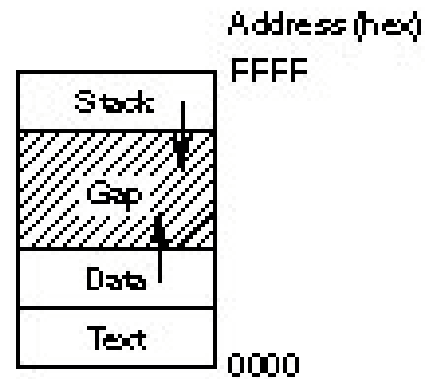




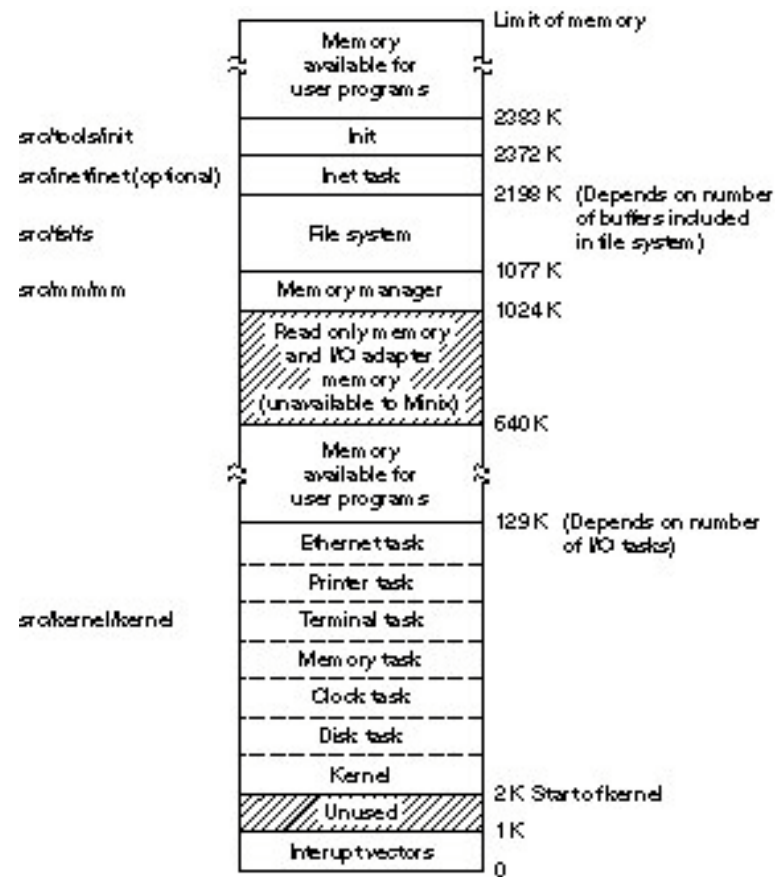
# Memória Primária e Dados

- O que contém a área de trabalho?
- Basicamente:
  - Instruções
  - Dados estáticos
  - Dados dinâmicos
    - Informações de controle (ex. controle de laços e endereços de retornos de subrotinas)
    - Dados de trabalho que são alocados e desalocados dinamicamente
- Isto quer dizer que o tamanho da área de trabalho não é conhecido de antemão

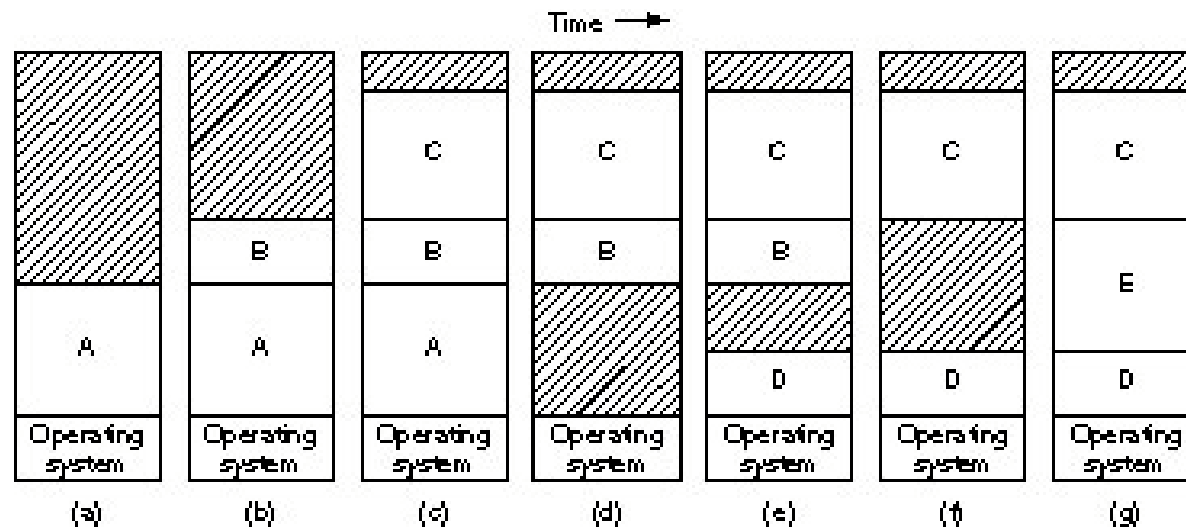
# Memória Primária e Dados



# Memória Primária e Dados



# Memória Primária e Dados







# Memória Primária e Dados

- Localidade de referência:
  - Propriedade importante de programas onde estatisticamente há uma probabilidade maior de buscar um dado próximo do que um dado distante



# Memória Primária e Dados

## ■ Cenário típico:

- Processo X vai iniciar e precisa de área de memória
- Sistema operacional (*memory manager*) consulta tabelas para ver que áreas da RAM podem ser alocadas
- Aloca área da RAM para o processo X
- Processo X inicia usando dados e alocando espaço
- [Gerenciador de memória sempre ativo]
- Se falta memória ou se algum outro processo precisa de mais área, algum processo deve ceder espaço, por exemplo, X
- Parte do espaço de X fica em disco então



# Memória Primária e Dados

## ■ Observações:

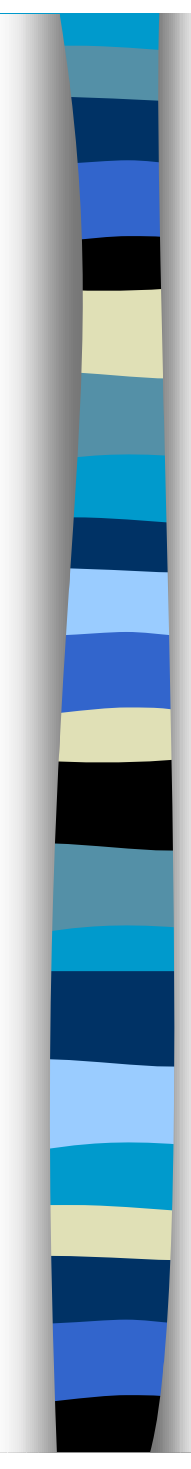
- Necessária uma política de escolha de áreas para sair
- Endereçamento da área de um processo deve ser independente da posição física e do dispositivo de base (endereço virtual) - em geral resolvido por hardware (MMU - Memory Management Unit)
- Processos bloqueados provavelmente terão seus espaços descarregados em disco



# Memória Primária e Dados

## ■ Implementação:

- Física: MMU + RAM particionada em páginas de tamanho fixo + disco (organização lógica em arquivos)
- Lógica:
  - Tabela de alocação de páginas físicas para processos
  - Algoritmos de substituição de páginas



```

last pid: 2661; load averages: 0.00, 0.01, 0.01
17:54:55
105 processes: 103 sleeping, 1 zombie, 1 on cpu
CPU states: % idle, % user, % kernel, % iowait, %
swap
Memory: 4096M real, 1376M free, 81M swap in use, 939M swap free

```

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	TIME	CPU	COMMAND
2661	jpg	1	0	0	1608K	1472K	cpu1	0:00	0.38%	top
399	root	1	58	0	2272K	1392K	sleep	10:41	0.04%	_upsd
400	root	2	59	-15	4464K	2880K	sleep	28:30	0.00%	nsrmmmd
809	root	1	33	0	3120K	2248K	sleep	16:57	0.00%	sshd2
521	root	1	59	0	31M	28M	sleep	11:00	0.00%	Xsun
2911	setubal	1	48	0	40M	30M	sleep	7:17	0.00%	netscape
398	root	1	28	10	4424K	2840K	sleep	6:44	0.00%	nsrindexd
197	root	6	58	0	2824K	2024K	sleep	2:25	0.00%	automountd
102	root	9	59	0	6616K	4424K	sleep	1:44	0.00%	named
1029	setubal	1	59	0	17M	11M	sleep	1:27	0.00%	emacs
125	root	1	58	0	1968K	1392K	sleep	0:49	0.00%	ypserv
439	root	2	59	-15	3704K	2672K	sleep	0:34	0.00%	nsrmmmd
17854	katsumi	1	46	4	20M	16M	sleep	0:27	0.00%	netscape
323	root	1	58	0	4616K	3392K	sleep	0:17	0.00%	nsrd
5995	katsumi	4	47	4	19M	14M	sleep	0:14	0.00%	acroread

jpk@onsona% ps -elf | more

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	CMD
19	T	root	0	0	0	0	SY	10417f10	0		Mar 04 ?		0:28	sched
8	S	root	1	0	0	41	20	629a5608	88	629a5800	Mar 04 ?		0:02	/etc/init -
19	S	root	2	0	0	0	SY	629a4f48	0	104396f8	Mar 04 ?		0:00	pageout
19	S	root	3	0	1	0	SY	629a4888	0	1043c934	Mar 04 ?		208:18	fsflush
8	S	root	319	1	0	40	8	62bb9610	241	62bb9808	Mar 04 ?		0:00	/usr/lib/inet/xntpd
8	S	root	102	1	0	40	20	629a26c8	827	ef4e1e7c	Mar 04 ?		1:44	/usr/local/net/sbin/named -c /etc/n
8	S	root	522	485	0	59	20	630b0030	769	630b00a0	Mar 04 ?		0:00	/usr/dt/bin/dtlogin -daemon
8	S	root	158	1	0	41	20	62bb7b10	268	62a6ba3e	Mar 04 ?		0:01	/usr/sbin/inetd -s
8	S	root	114	1	0	41	20	62bb81d0	253	62a6bd0e	Mar 04 ?		0:00	/usr/sbin/keyserv
8	S	daemon	164	1	0	41	20	62bb66d0	288	ef30de7c	Mar 04 ?		0:00	/usr/lib/nfs/statd
8	S	root	125	1	0	41	20	62c70f58	246	62a6bc96	Mar 04 ?		0:49	/usr/lib/netsvc/yp/ypserv -d
8	S	root	112	1	0	41	20	62bb8f50	264	62a6bdd6	Mar 04 ?		0:08	/usr/sbin/rpcbind
8	S	root	127	125	0	41	20	629a2d88	198	62a6bc6e	Mar 04 ?		0:01	rpc.nisd_resolv -F -C 8 -p 10737418
8	S	root	135	1	0	55	20	62c6f458	226	62a6bbce	Mar 04 ?		0:00	/usr/lib/netsvc/yp/ypxfrd
8	S	root	132	1	0	41	20	62c6fb18	237	62a6bba6	Mar 04 ?		0:00	/usr/lib/netsvc/yp/ypbind
8	S	root	321	1	0	41	20	62f3a6e8	276	62a6b3d6	Mar 04 ?		0:03	/usr/lib/sendmail -C /etc/mail/send
8	S	root	142	1	0	95	20	62c6e6d8	205	62a6bab6	Mar 04 ?		0:00	/usr/lib/netsvc/yp/rpc.yppupdated
8	S	root	140	1	0	85	20	629a41c8	220	62a6bb56	Mar 04 ?		0:00	/usr/lib/netsvc/yp/rpc.yppasswdd -m
8	S	setubal	521	485	0	40	20	62bb7450	4020	62a6b4ee	Mar 04 ?		11:00	/usr/openwin/bin/Xsun :0 -nobanner
8	S	root	175	1	0	41	20	629a3448	223	62a6b9ee	Mar 04 ?		0:00	/usr/lib/nfs/lockd
8	S	root	197	1	0	41	20	629a2008	353	629a26b0	Mar 04 ?		2:25	/usr/lib/autofs/automountd
8	S	root	201	1	0	41	20	62c6e018	597	eef05e7c	Mar 04 ?		0:04	/usr/sbin/syslogd
8	S	root	300	1	0	41	20	62f3c1e8	297	62a6af26	Mar 04 ?		0:01	/usr/sbin/vold
8	S	root	240	1	0	47	20	62daaf60	308	62dab158	Mar 04 ?		0:05	/usr/sbin/nsd
8	S	root	234	1	0	51	20	62c70898	204	6023deb8	Mar 04 ?		0:00	/usr/sbin/cron
8	S	daemon	348	347	0	40	20	62bb8890	812	62a6b106	Mar 04 ?		0:00	/usr/lib/ab2/dweb/sunos5/bin/dwhttp
8	S	root	492	1	0	41	20	631aef78	236	62a6a7f6	Mar 04 ?		0:00	/usr/lib/snmp/snmpdx -y -c /etc/snm
8	S	root	809	389	0	66	20	633b5488	390	62a6a27e	Mar 04 ?		16:58	/usr/local/net/sbin/sshd
8	S	root	349	323	0	71	30	62daa1e0	508	62a6b336	Mar 04 ?		0:06	/usr/sbin/nsrmmdbd
8	S	root	255	1	0	41	20	62bb6010	336	62a6b926	Mar 04 ?		0:00	/usr/lib/lpsched
8	S	root	439	323	0	40	5	62bb6d90	463	62a6abb6	Mar 04 ?		0:35	/usr/sbin/nsrmmmd -n 2
8	S	root	291	1	0	41	20	62f3cf68	116	62a6b386	Mar 04 ?		0:00	/usr/lib/utmpd
8	S	root	323	1	0	41	20	62da8da0	577	62a6ac2e	Mar 04 ?		0:17	/usr/sbin/nsrd
8	S	root	280	1	0	49	20	62da86e0	152	62da88d8	Mar 04 ?		0:00	/usr/lib/power/powerd
8	S	root	356	1	0	99	20	62c6ed98	122	62c6ee08	Mar 04 ?		0:00	/bin/sh /usr/local/bin/safe_mysqlld
8	S	root	310	1	0	41	20	62f3c8a8	304	62a6b30e	Mar 04 ?		0:01	/usr/sbin/nsrexecd
8	S	root	314	310	0	41	20	62f3b468	326	62a6b156	Mar 04 ?		0:01	/usr/sbin/nsrexecd

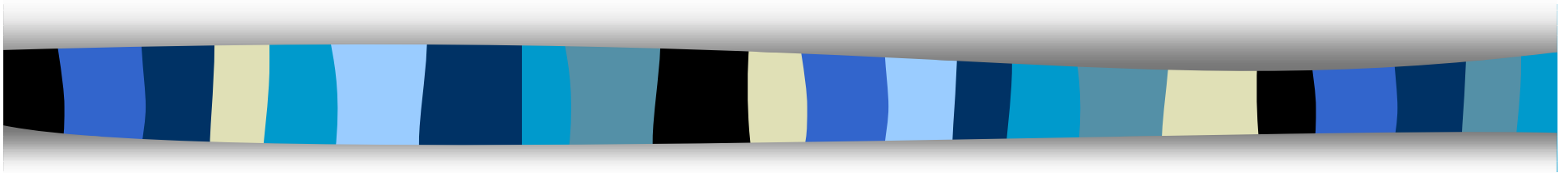


# Memória Primária e Dados

## ■ Aspectos adicionais:

- Segurança: espaços devem ser protegidos
- Estatísticas de acesso a páginas são importantes: páginas muito acessadas devem ser mantidas em RAM

# Memória Secundária e Arquivos







# Memória Secundária e Arquivos

- Arquivos são abstrações relacionadas com armazenamento:
  - Não perene (diferente da área de trabalho de um processo)
  - “Estático”
  - Número de arquivos em geral muito maior do que número de processos e por conseguinte ocupam uma área maior



# Memória Secundária e Arquivos

- Arquivos ocupam um espaço contíguo de endereçamento lógico (um arquivo pode estar fisicamente distribuído em espaços não contíguos)
- Arquivos contêm:
  - Dados numéricos, textuais ou arquivos binários (nem numérico, nem textual)



# Memória Secundária e Arquivos

## ■ Atributos de arquivos:

- Nome
- Tipo
- Localização
- Tamanho
- Proteção
- Identificação de usuário, tempo e data
- Informações de arquivos são armazenadas na estrutura *diretório*

# Memória Secundária e Arquivos

```
jpk@araguaia% ls -al
total 264
drwx--x--x  8 jpk      spec      4096 Mar 11 23:30 .
drwxr-xr-x 111 root    root      4096 Mar 11 04:07 ..
-rw-----  1 jpk      spec      1432 Mar  5 08:49 .Xauthority
-rw-r--r--  1 jpk      spec        653 Feb  5 08:08 .addressbook
-rw-----  1 jpk      spec     2684 Feb  5 08:08 .addressbook.lu
-rw-----  1 jpk      spec    3471 Mar 11 00:49 .bash_history
-rw-r--r--  1 jpk      spec      956 Oct  5 12:21 .desksetdefaults
drwx-----  3 jpk      spec     4096 Feb  5  2001 .fm
-rwxr-xr-x  1 jpk      spec     6755 Feb  5  2001 .login
drwx-----  4 jpk      spec     4096 Oct 30 14:41 .netscape
-rw-----  1 jpk      spec     9432 Mar 11 23:56 .pine-debug1
-rw-----  1 jpk      spec     8366 Mar 11 22:21 .pine-debug2
-rw-----  1 jpk      spec     7848 Mar 11 22:20 .pine-debug3
-rw-----  1 jpk      spec    14214 Mar 11 22:20 .pine-debug4
-rw-r--r--  1 jpk      spec    11595 Mar  1 07:07 .pinerc
-rwxr-xr-x  1 jpk      spec     8208 Jan  2 08:47 .profile
drwxr-xr-x  2 jpk      spec     4096 Sep 25  2000 .ssh
drwxr-xr-x  3 jpk      spec     4096 Dec 29 11:43 .ssh2
drwxr-xr-x  2 jpk      spec     4096 Feb  5  2001 .wastebasket
drwx-----  2 jpk      spec     4096 Mar 11 23:56 mail
jpk@araguaia%
```



# Memória Secundária e Arquivos

## ■ Operações sobre arquivos

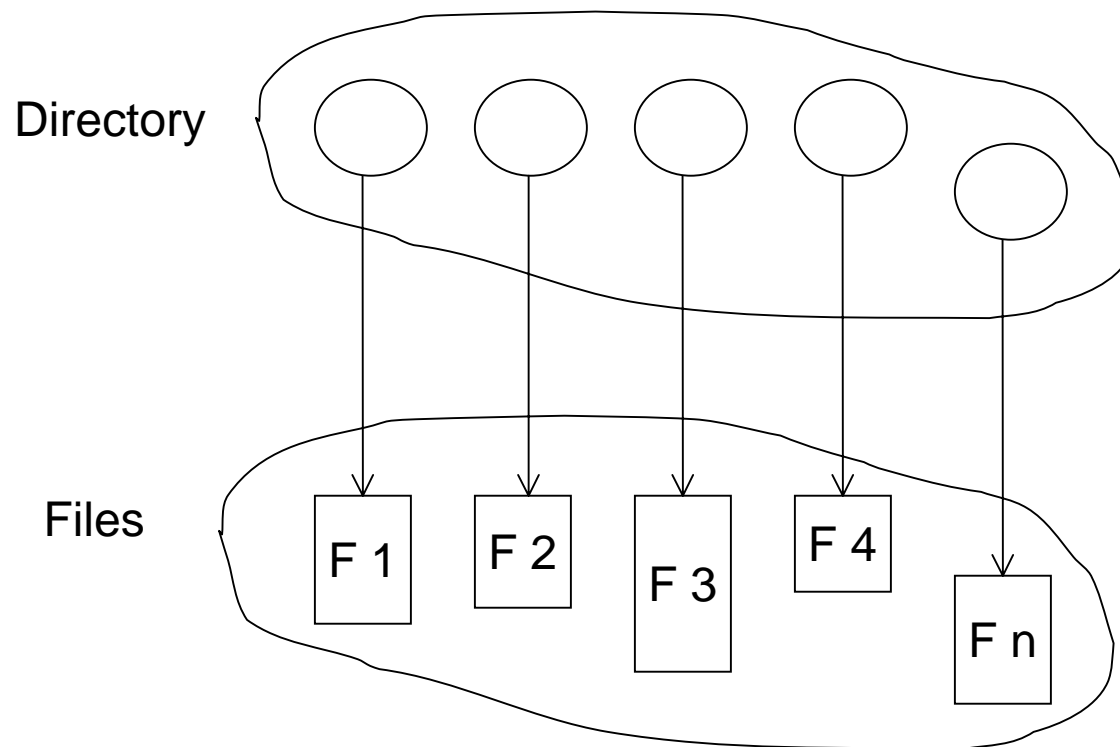
- Criação
- Escrita
- Leitura
- Deslocamento dentro do arquivo (*seek*)
- Remoção
  
- Open: mover conteúdo do arquivo em disco para memória
- Close: mover conteúdo do arquivo em memória para disco

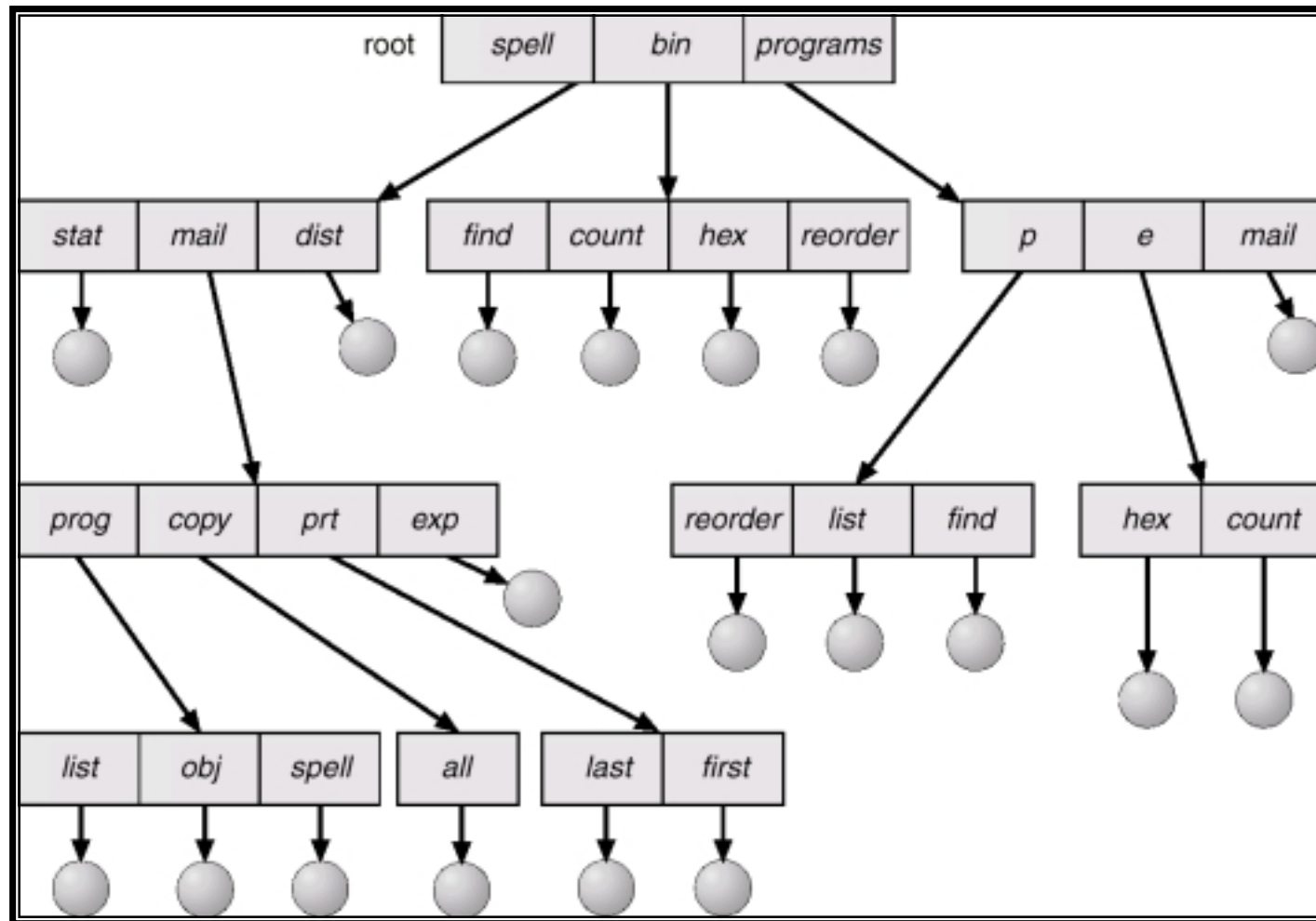


# Memória Secundária e Arquivos

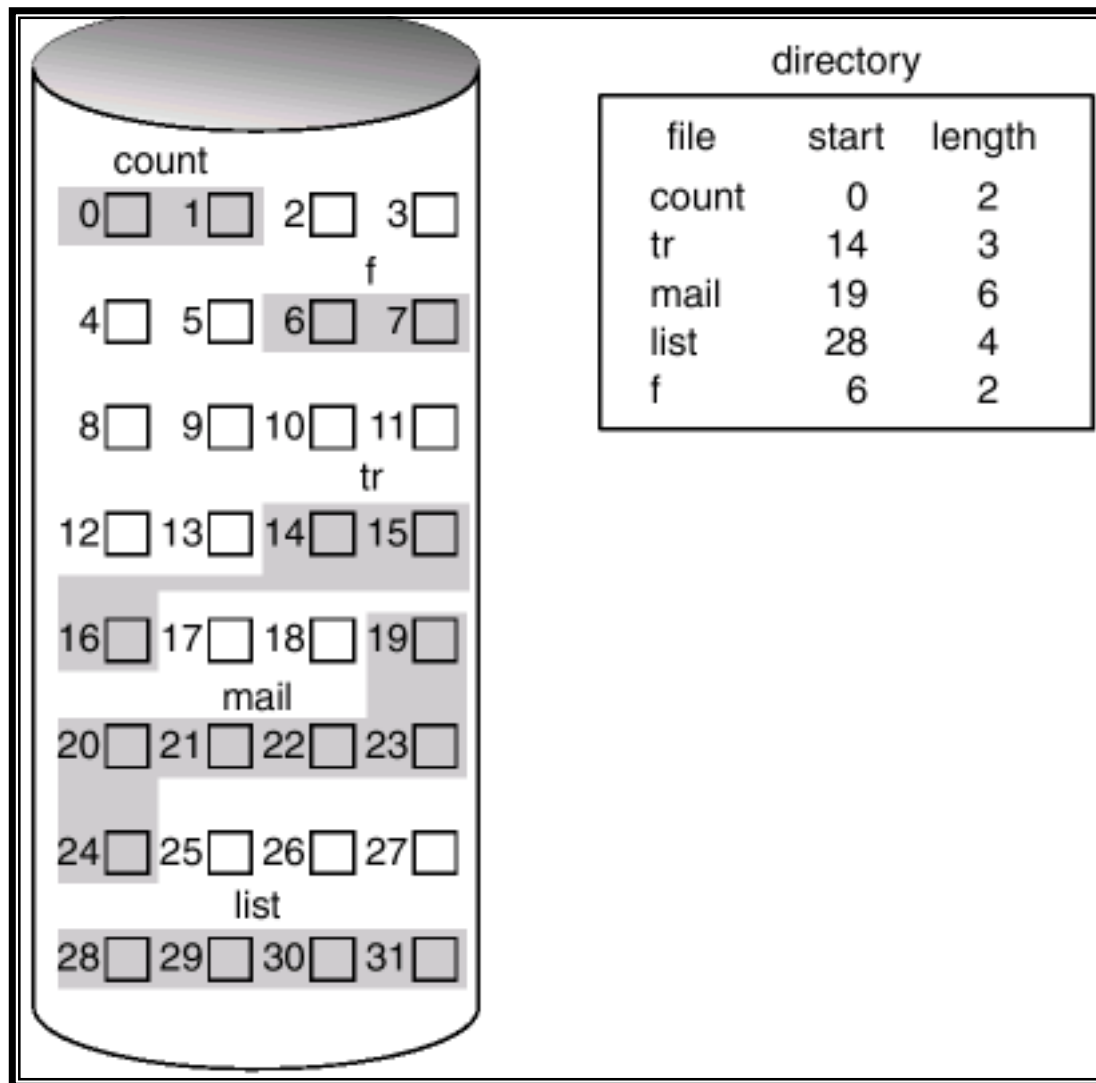
file type	usual extension	function
executable	exe, com, bin or none	read to run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

# Memória Secundária e Arquivos

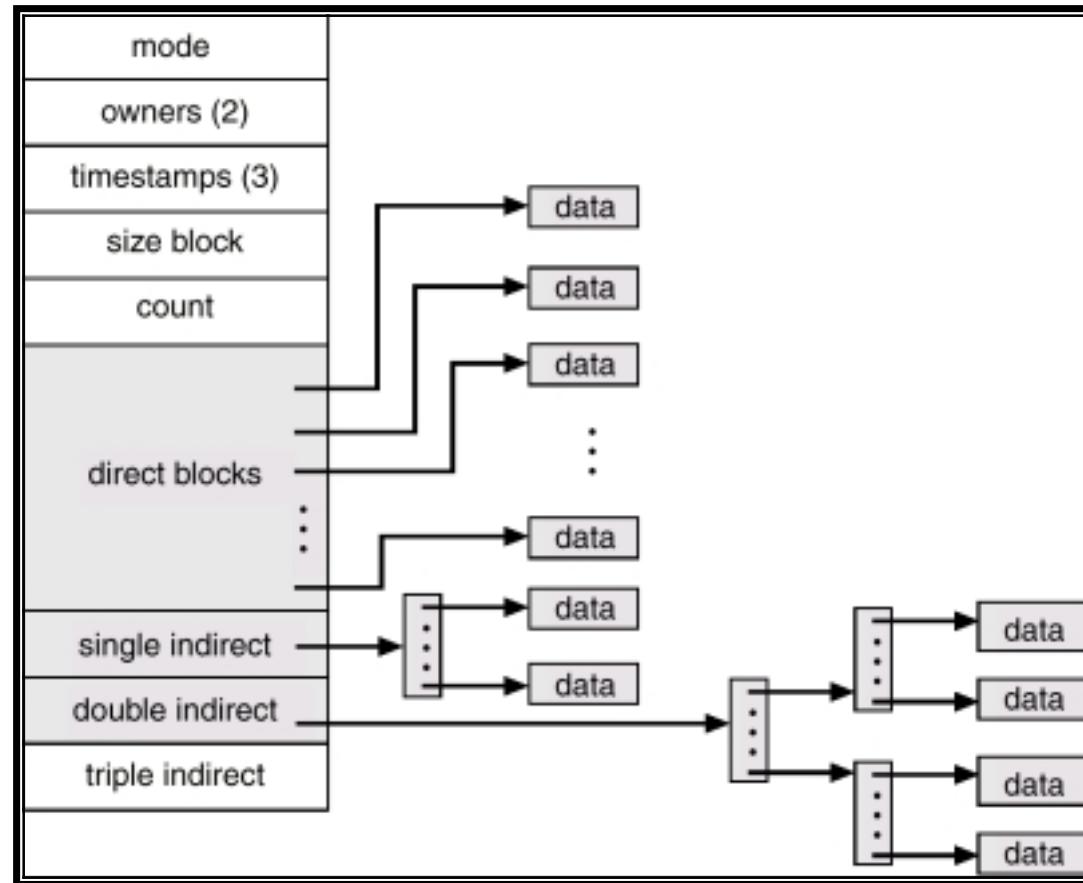




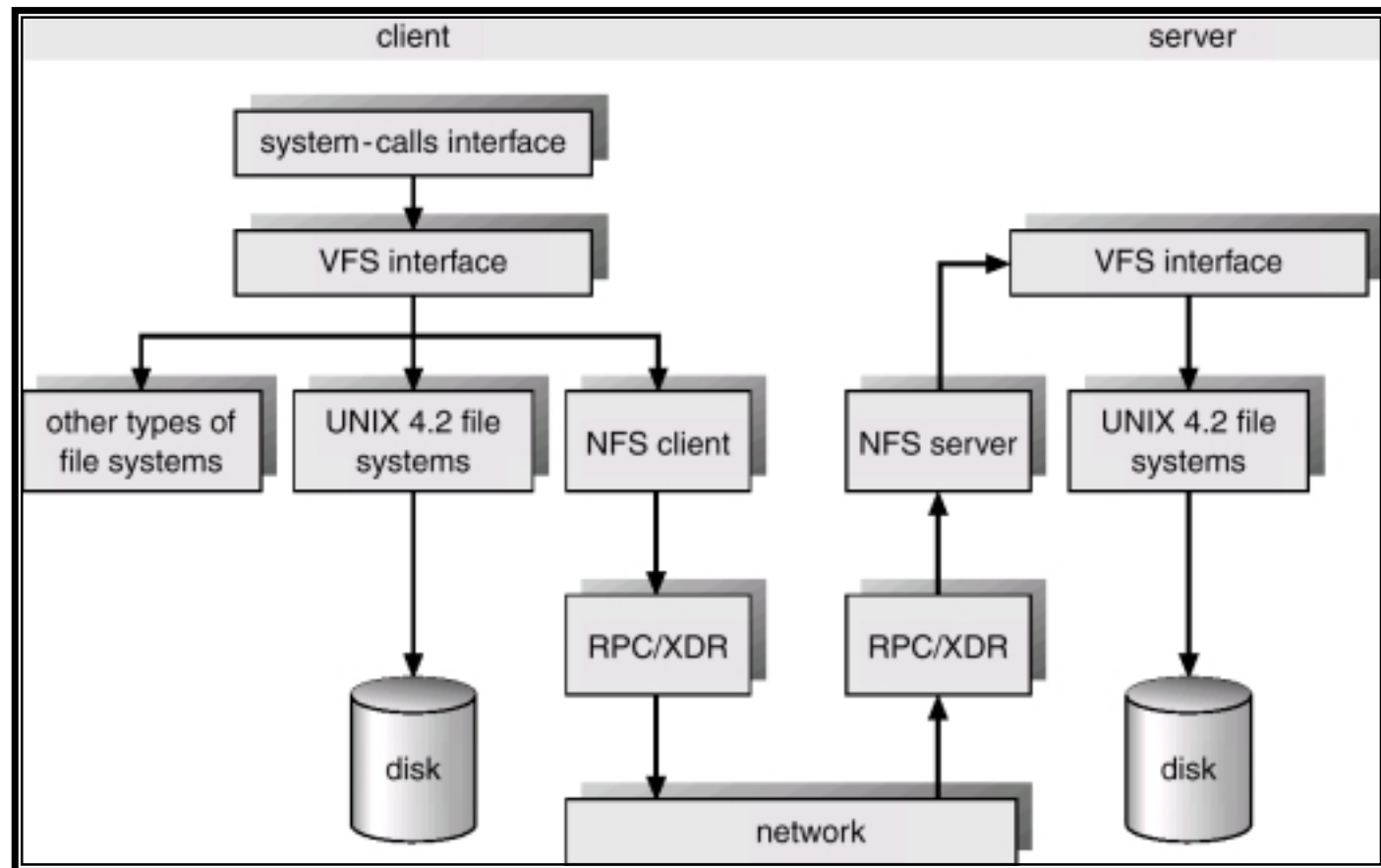




# Memória Secundária e Arquivos

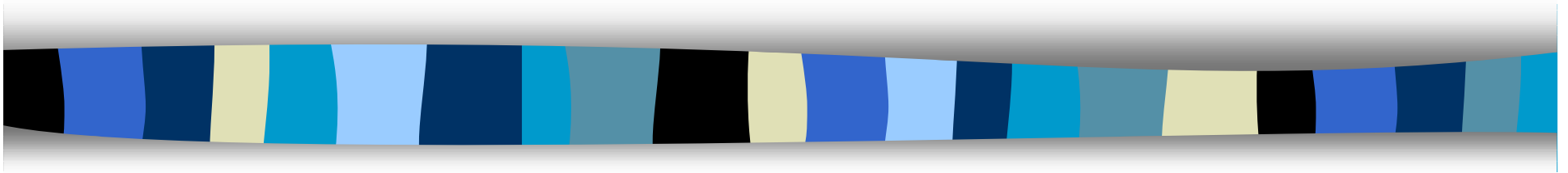


# Memória Secundária e Arquivos



# Periféricos e Entrada/Saída

(Tópico “Redes” incluído aqui)

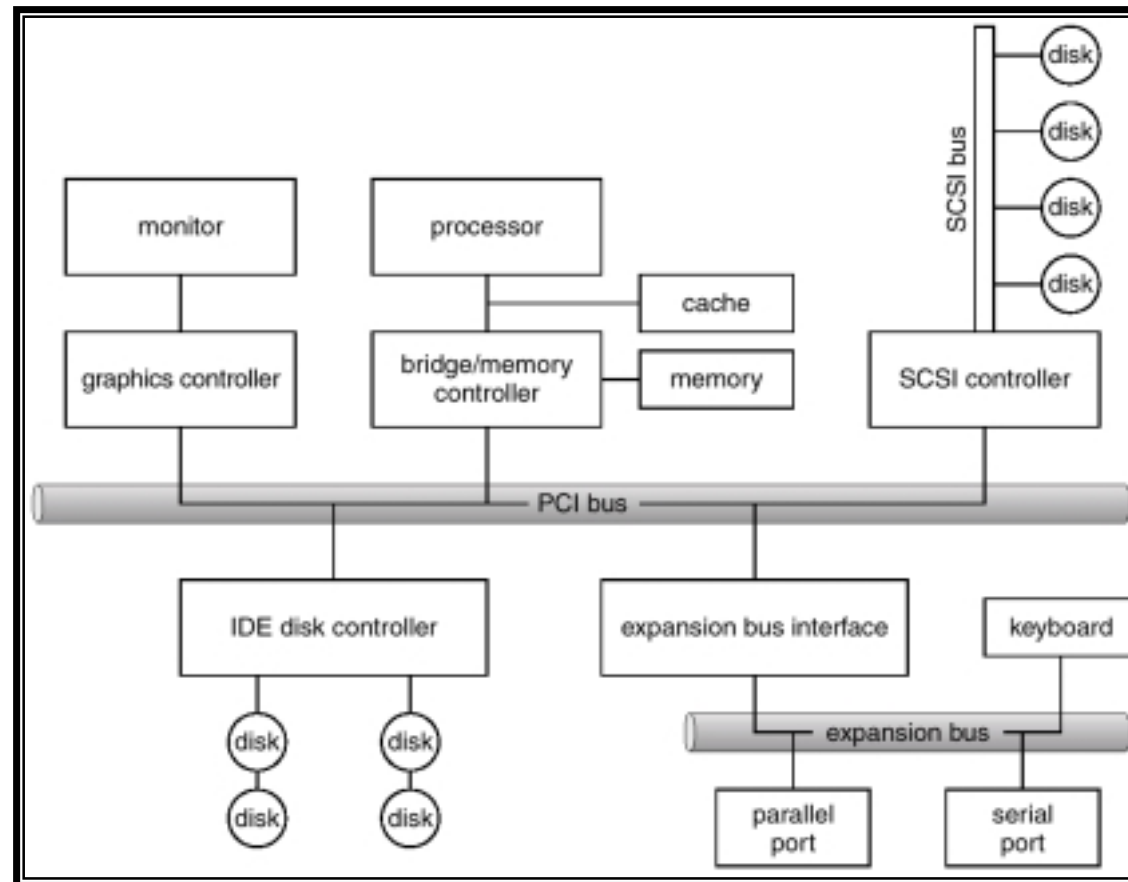




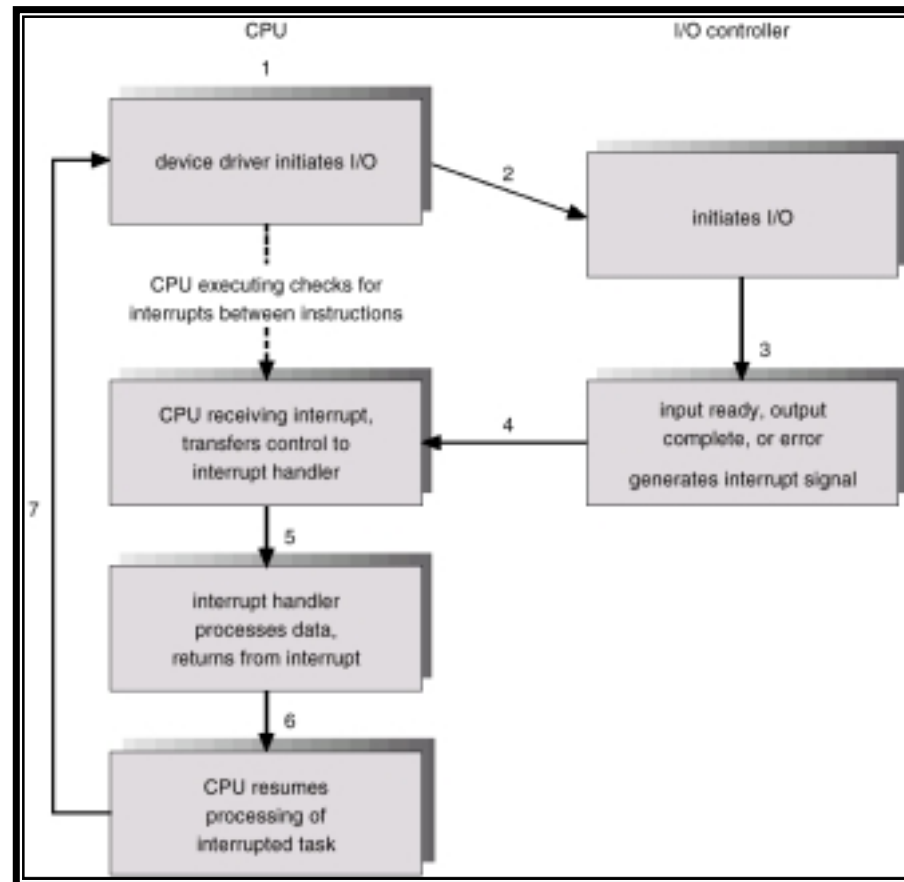
# Periféricos e Entrada/Saída

- Grande variedade de dispositivos de I/O
- Conceitos:
  - Porta
  - Barramento (Bus)
  - Controladora (host adapter)
  - Software do dispositivo: driver
- Instruções de E/S controlam os dispositivos
- Dispositivos têm endereços para uso pelas instruções e para o I/O mapeado na memória

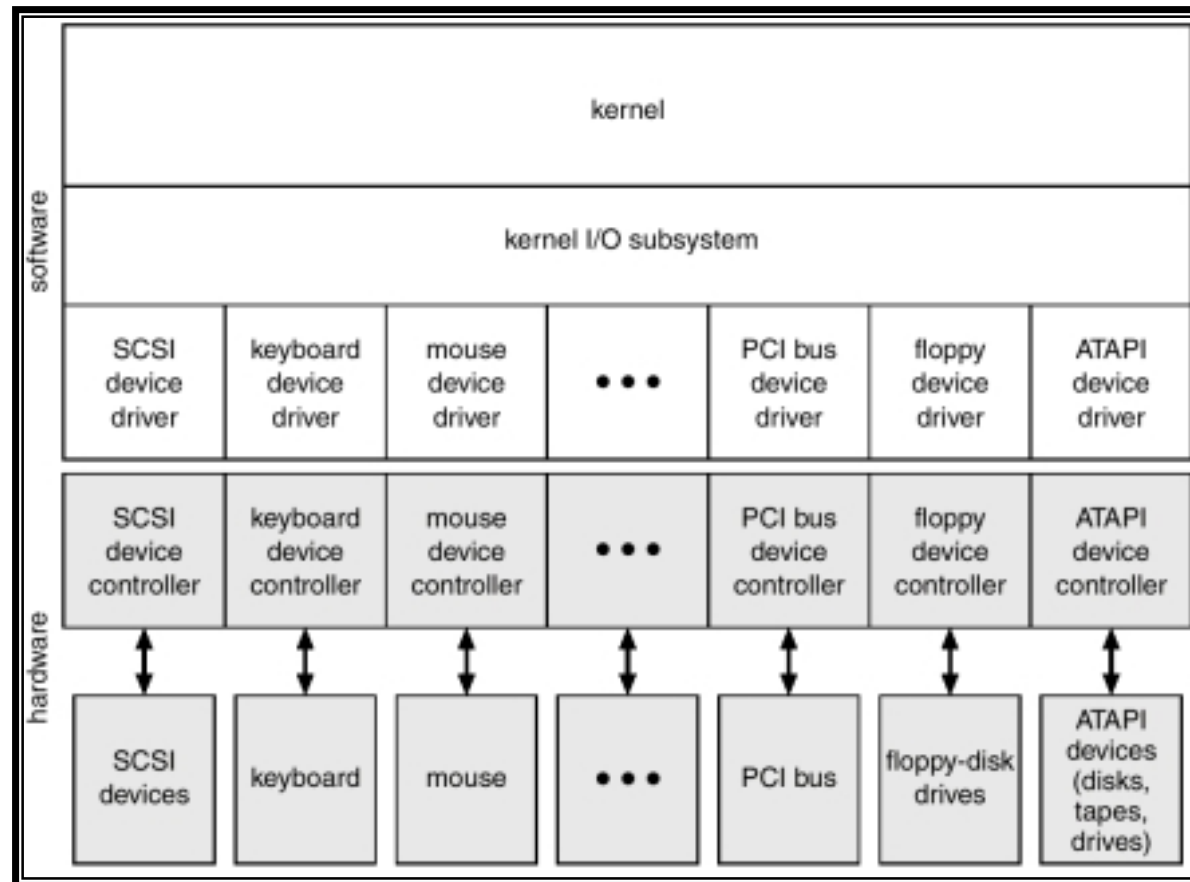
# Periféricos e Entrada/Saída



# Periféricos e Entrada/Saída



# Periféricos e Entrada/Saída

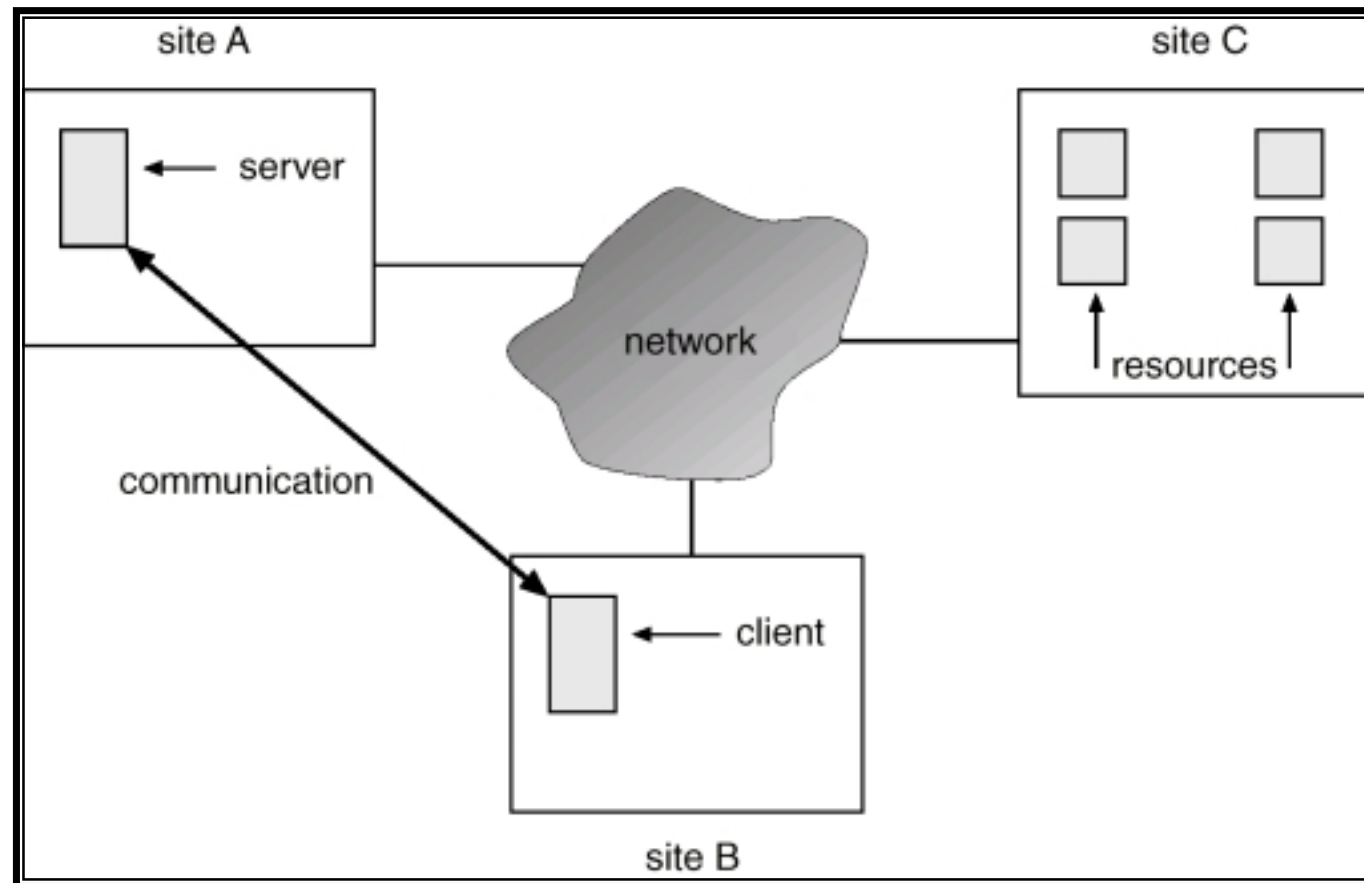




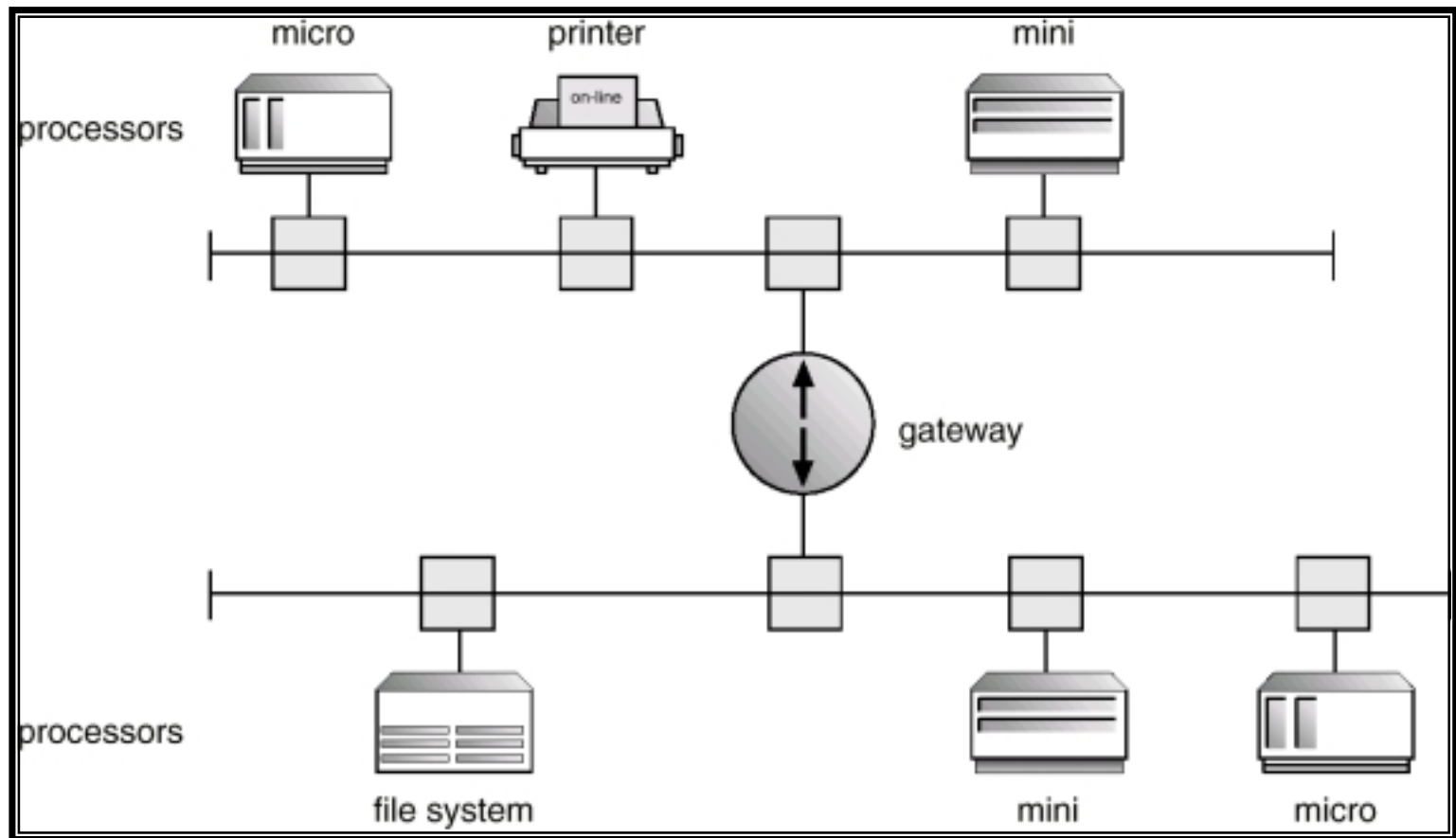
# Periféricos e Entrada/Saída

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk

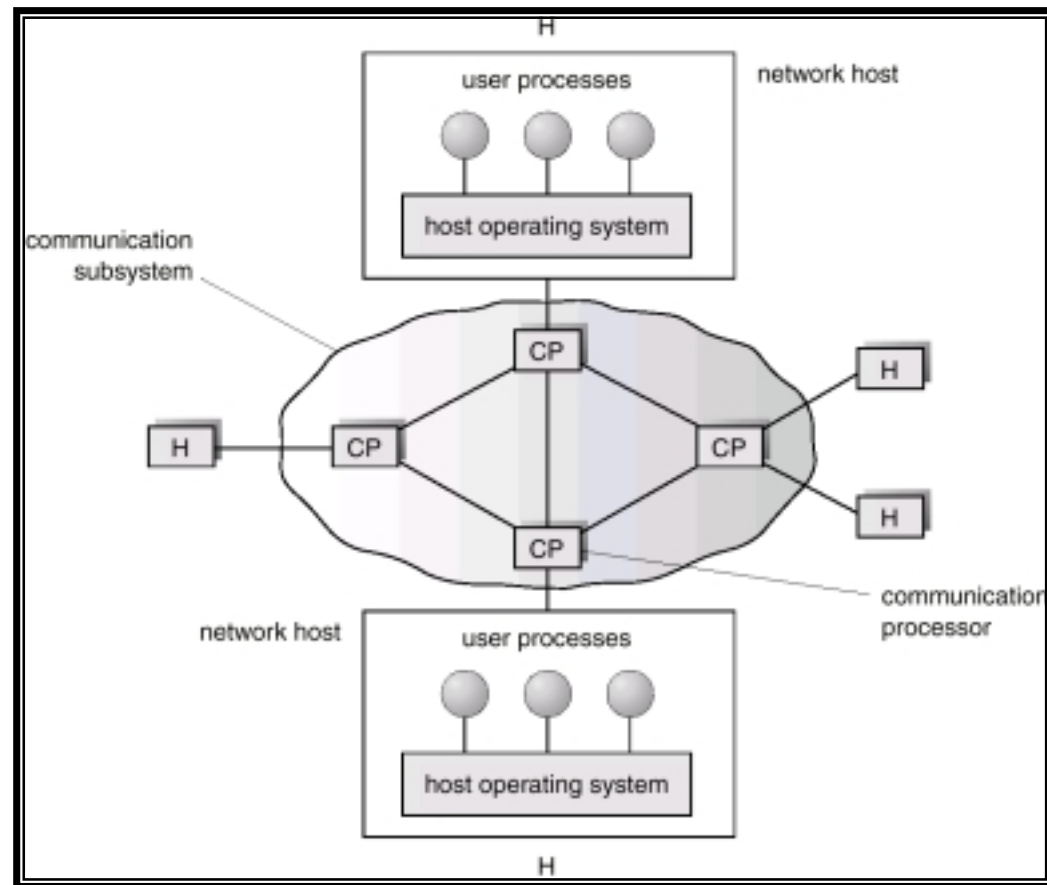
# Periféricos e Entrada/Saída



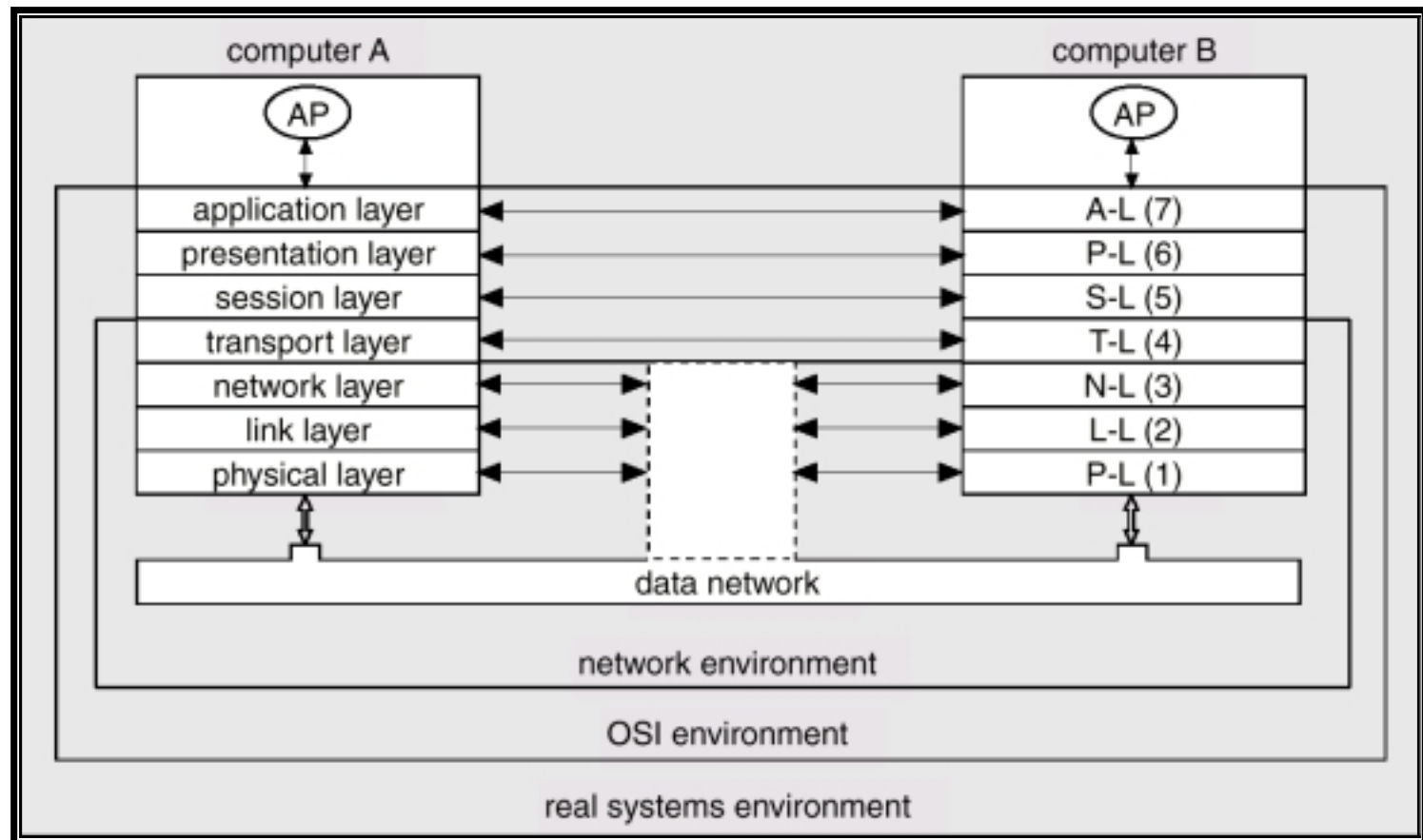
# Periféricos e Entrada/Saída

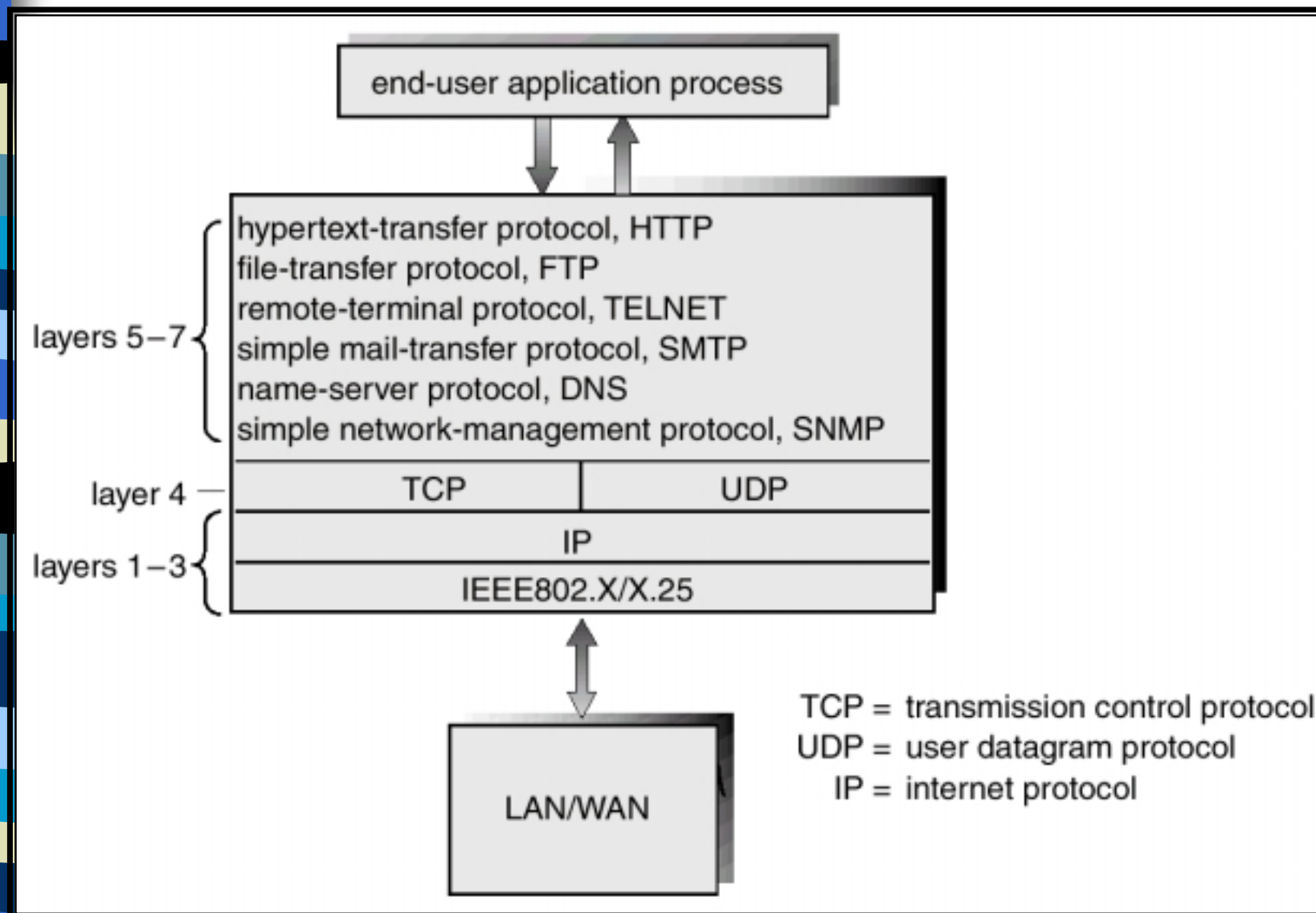


# Periféricos e Entrada/Saída

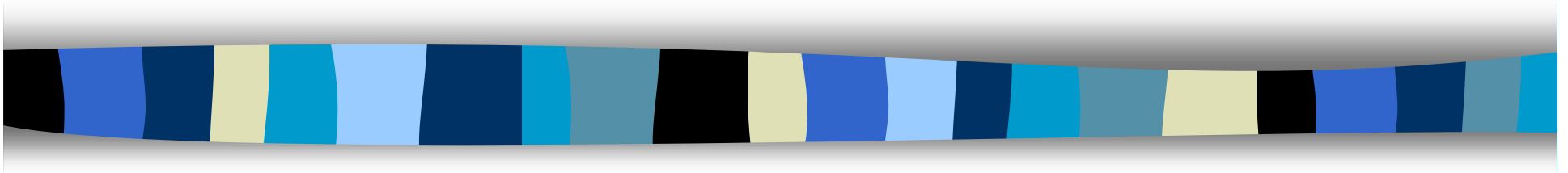


# Periféricos e Entrada/Saída





# Conclusões



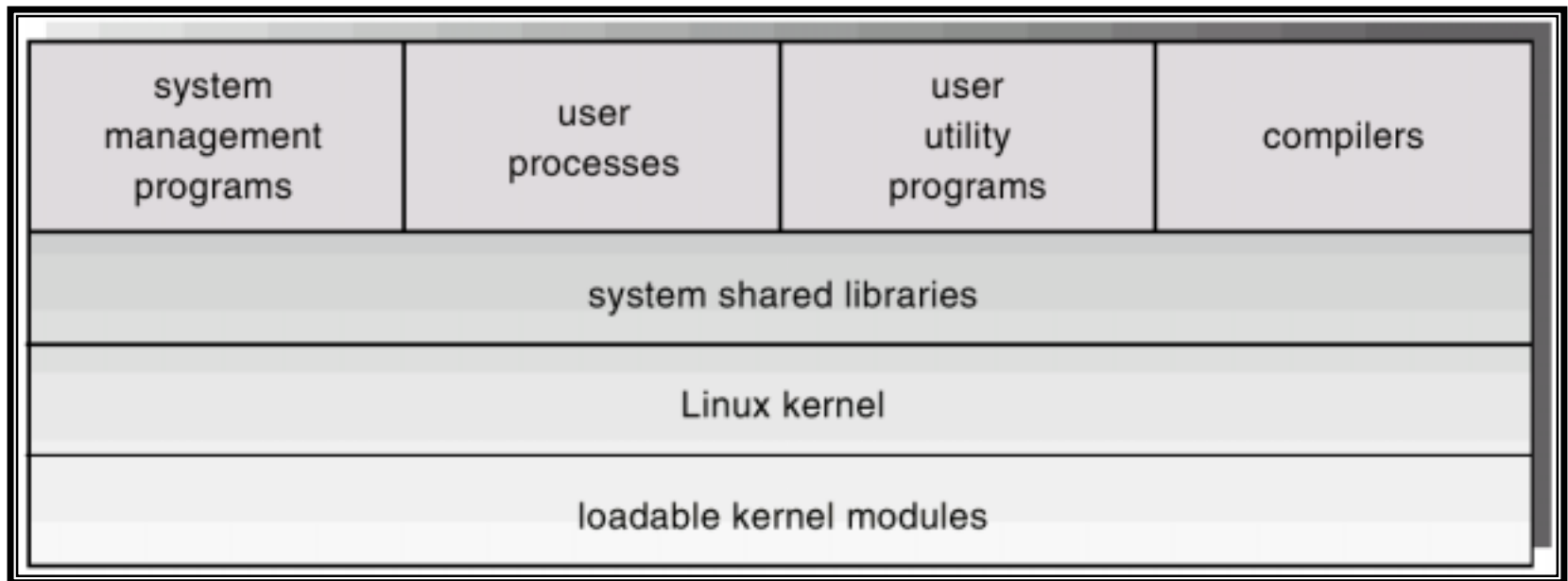


## Conclusões

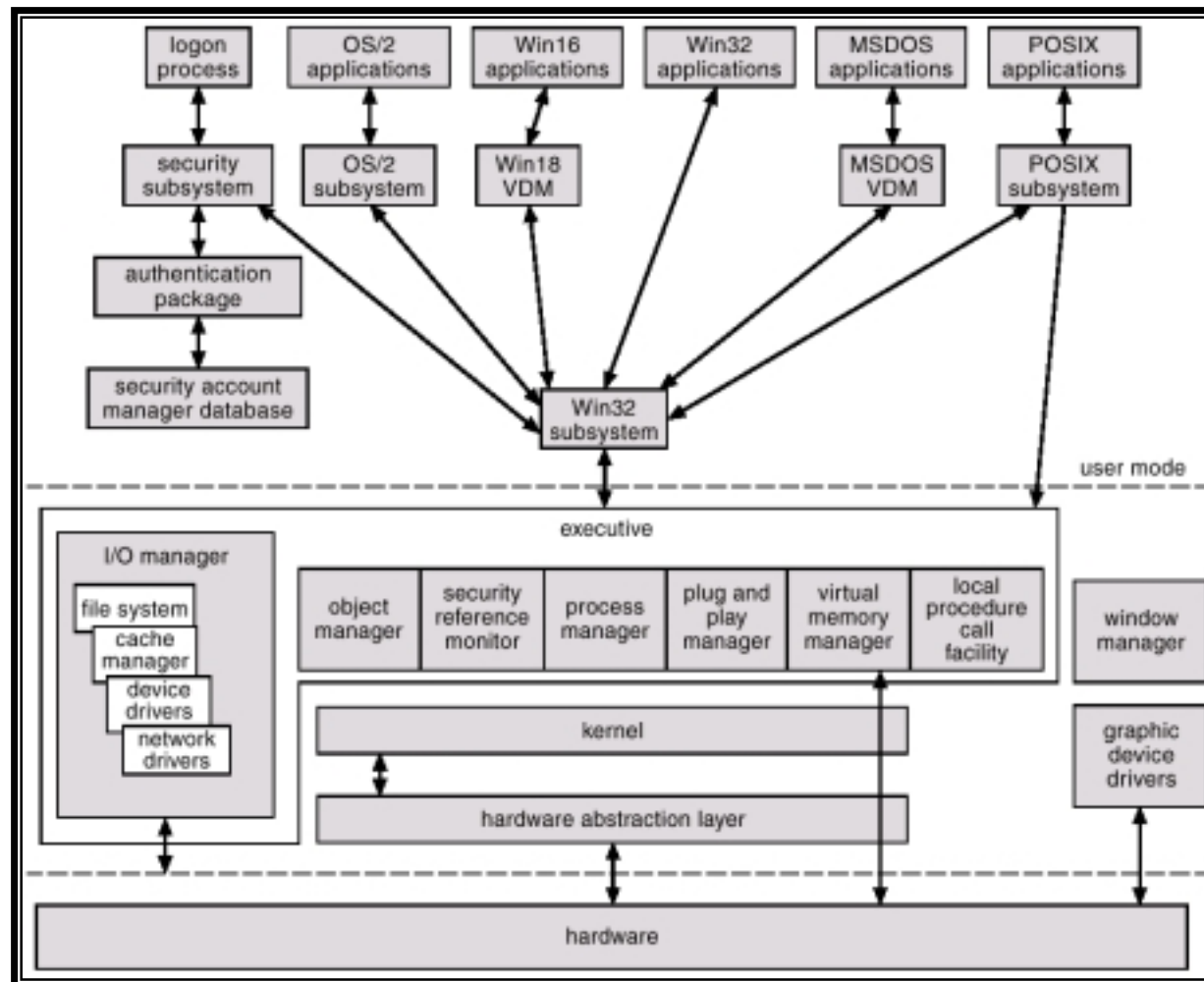
- Outros aspectos não cobertos aqui:
  - Proteção (acesso correto e por quem é autorizado)
  - Segurança (proteção contra ambiente externo)
  - Sistemas operacionais distribuídos
- Sistemas operacionais suportam conceitos de usuários com diferentes privilégios. Superusuário (*=root*): é em geral o administrador do sistema



# Conclusões



# Conclusões



# Conclusões



PC: Linux/Windows

Mac: MacOS



## Conclusões



Compaq: Tru64 Unix

Sun: V880 SunFire



The Sun Fire™ V880 server is ideal for remote offices and departments, and as a complementary system in data centers. Powered by UltraSPARC™ III technology, this rack-mountable desk-side unit excels in running database, e-commerce, and enterprise resource planning applications.

[Buy Online](#)

[Contact Me](#)

# Conclusões



HP: HP-UX

Silicon Graphics: Irix





# Conclusões

## ■ Referências:

- **Operating System Concepts Sixth Edition**  
[Abraham Silberschatz](#) [Peter Baer Galvin](#) [Greg Gagne](#).  
**John Wiley & Sons, Inc. ISBN 0-471-41743-2 Bound book**  
**date, June 15, 2001**
- **Operating Systems Design and Implementation**  
Second Edition by **Andrew S. Tanenbaum and Albert S.Woodhull**. **Prentice-Hall, 1999.**