

UFRJ – IM - DCC



Sistemas Operacionais I

Unidade I Estruturas de Sistemas Operacionais



ORGANIZAÇÃO DA UNIDADE

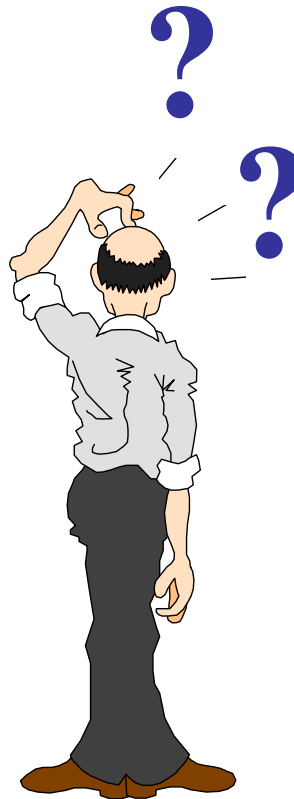
- Introdução
- Fundamentos de Hardware e Software
- Estruturas de Sistemas Operacionais
 - Componentes
 - Arquiteturas
 - Serviços
 - Estruturas de Controle
 - Chamadas ao SO



Estruturas do S.O. Definição e Objetivos

Programa que controla a execução de programas aplicativo

Interface entre o usuário e o hardware do computador



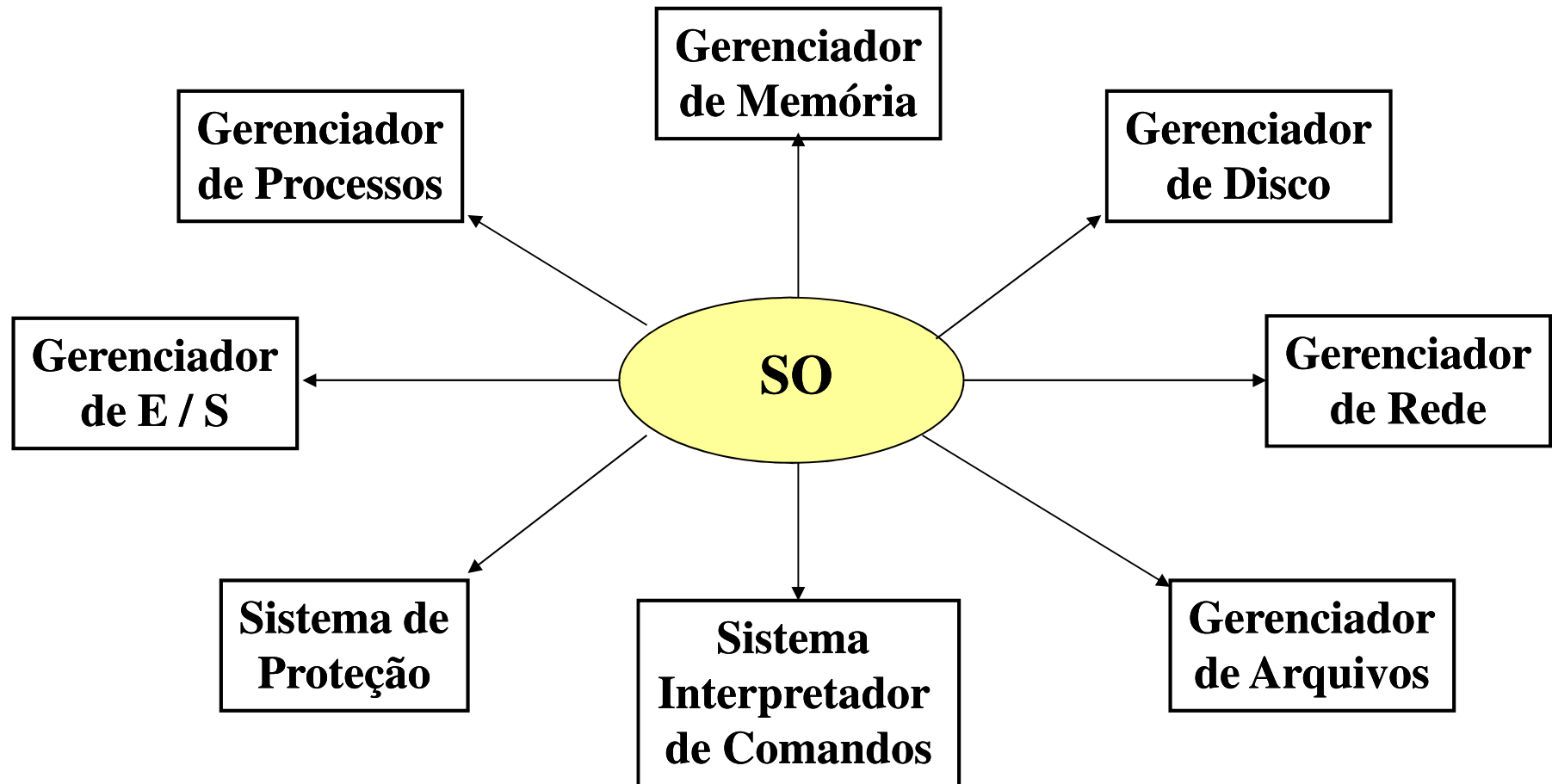
Torna o uso do computador mais conveniente

Permite um uso mais eficiente dos recursos

Passível de ter sua funcionalidade expandida transparentemente

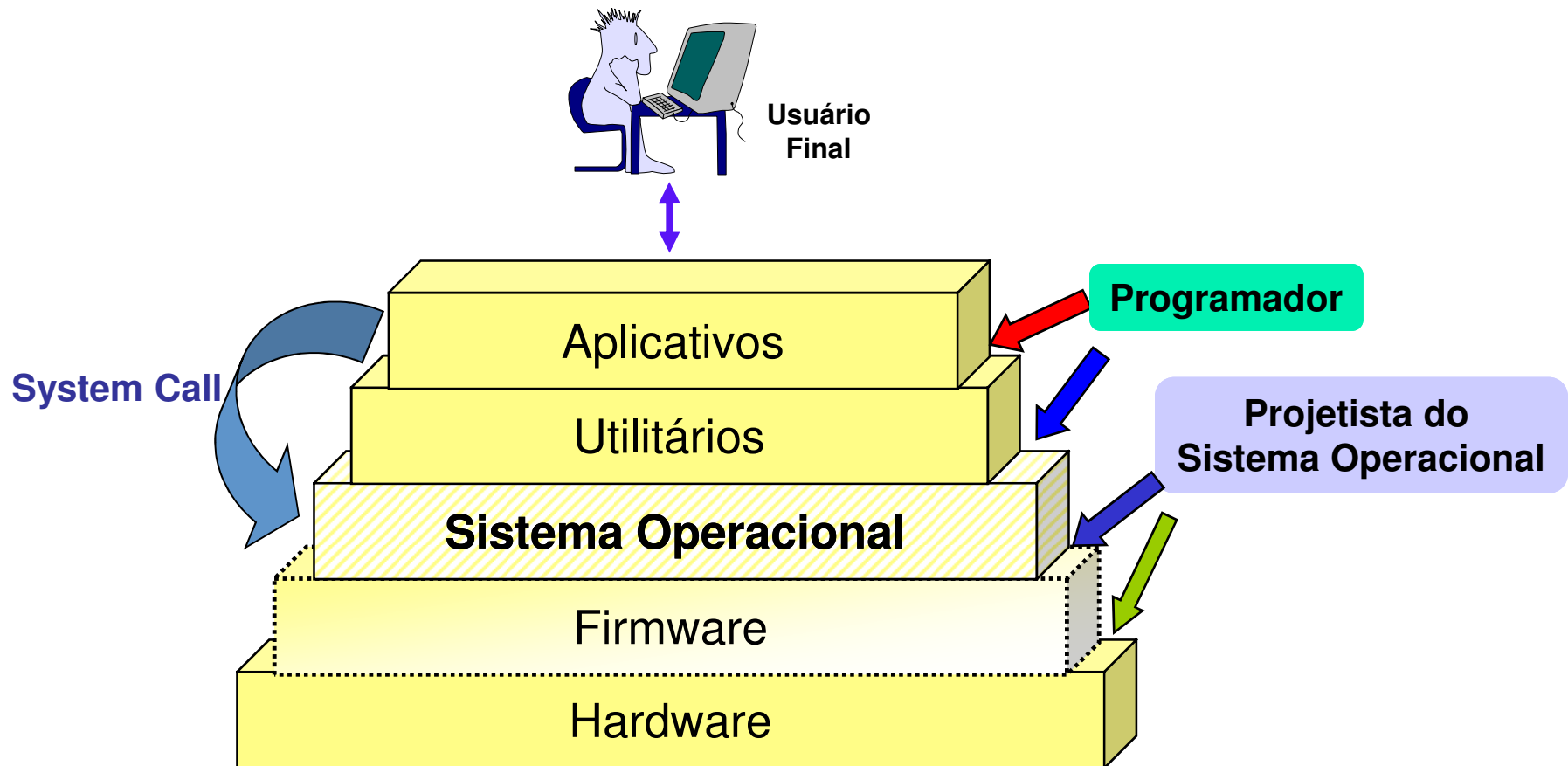


Estruturas do S.O. Componentes



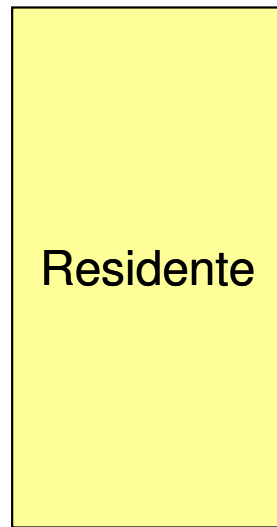


Estruturas do S.O. Posicionamento

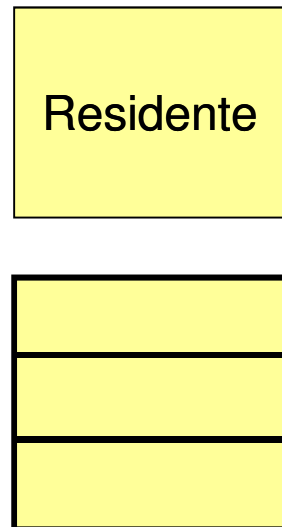




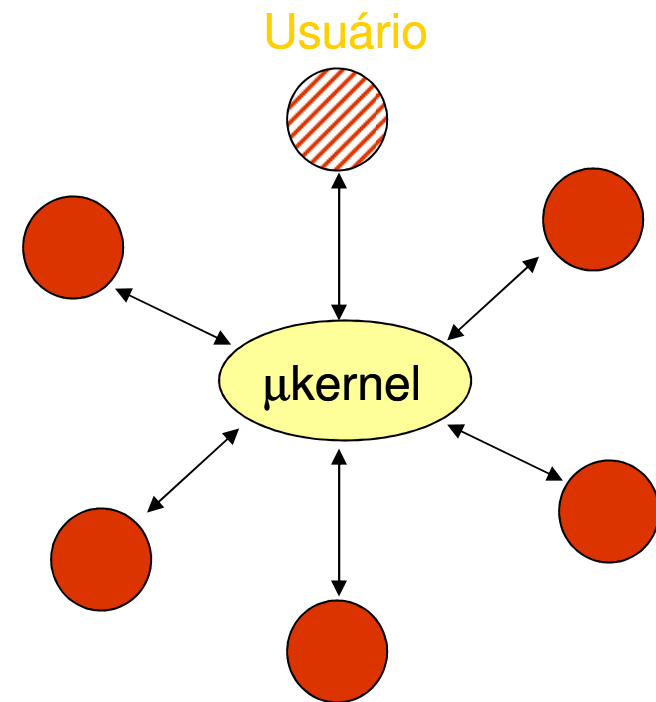
Estruturas do S.O. Arquiteturas



**Sistema
Monolítico**



**Sistema
Modular
(camadas)**



Cliente / Servidor



Estruturas do S.O. Arquitetura monolítica

Dominou até os primeiros grandes sistemas para Mainframes.

- CTSS (MIT, 1963) – 32.000 instruções de 36 bits cada
- OS/360 (IBM, 1964) – mais de 1 milhão de instruções
- MULTICS (MIT e Bell Labs) – mais de 20 milhões de instruções

Problemas:

- *Bugs*
- *Memória*
- *Complexidade*

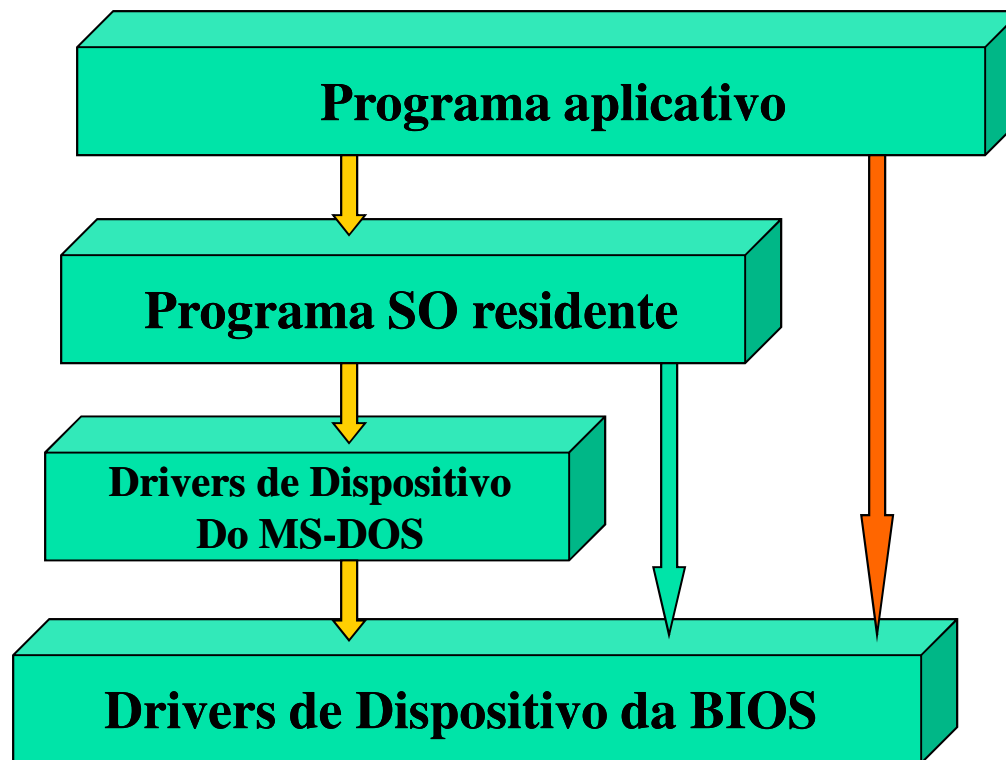


Arquitetura Modular e,
posteriormente, em
Camadas.



Estruturas do S.O. Arquitetura MS-DOS

Vários sistemas comerciais não têm uma estrutura definida, e o MS-DOS é um deles. No MS-DOS as interfaces e níveis de funcionalidade não são bem separados nem estão unificados

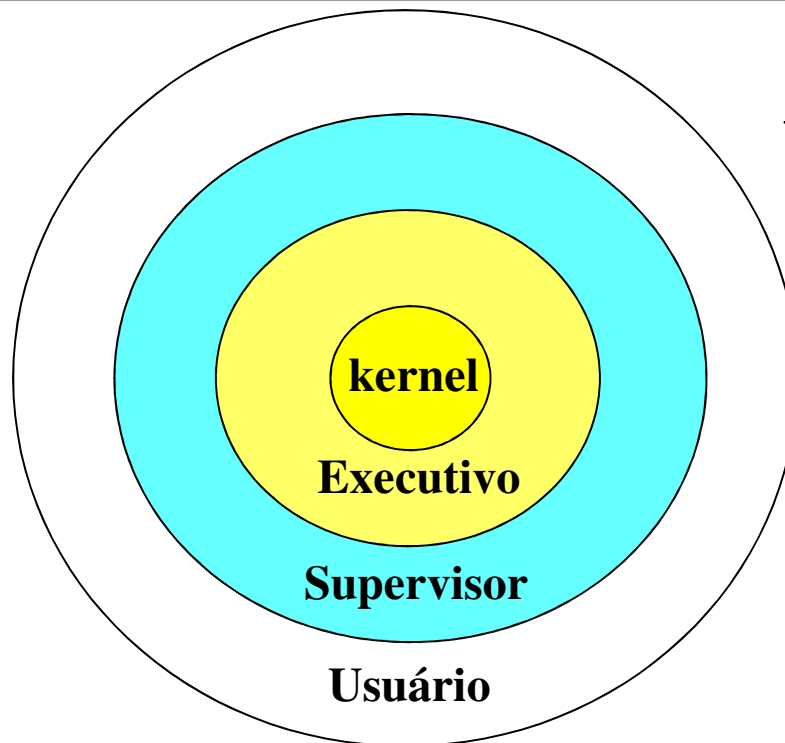


(*) O excesso de liberdade torna o SO vulnerável.



Estruturas do S.O. Arquitetura em camadas

O sistema é dividido em níveis sobrepostos. Cada nível oferece funções que só podem ser utilizadas pelas camadas mais externas.



Arquitetura do OpenVMS

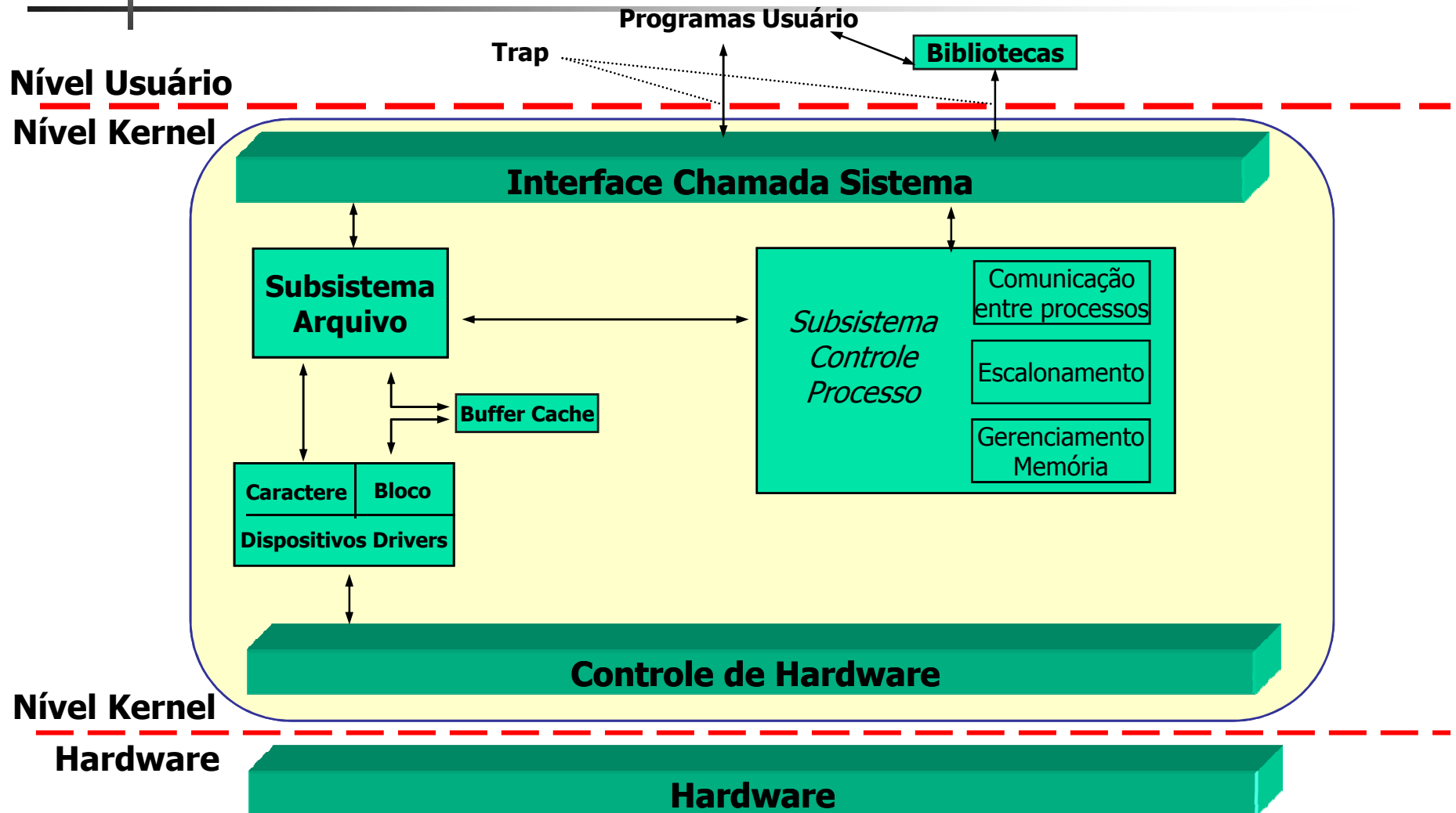
- THE (Technische Hogeschool Eindhoven – 1968) – primeiro em camadas (6)

- MULTICS e OpenVMS foram os seguintes, projetando-as concêntricas

- Atualmente a maioria adota 2 camadas (kernel – privilegiado e usuário – não privilegiado) – UNIX e Win2000

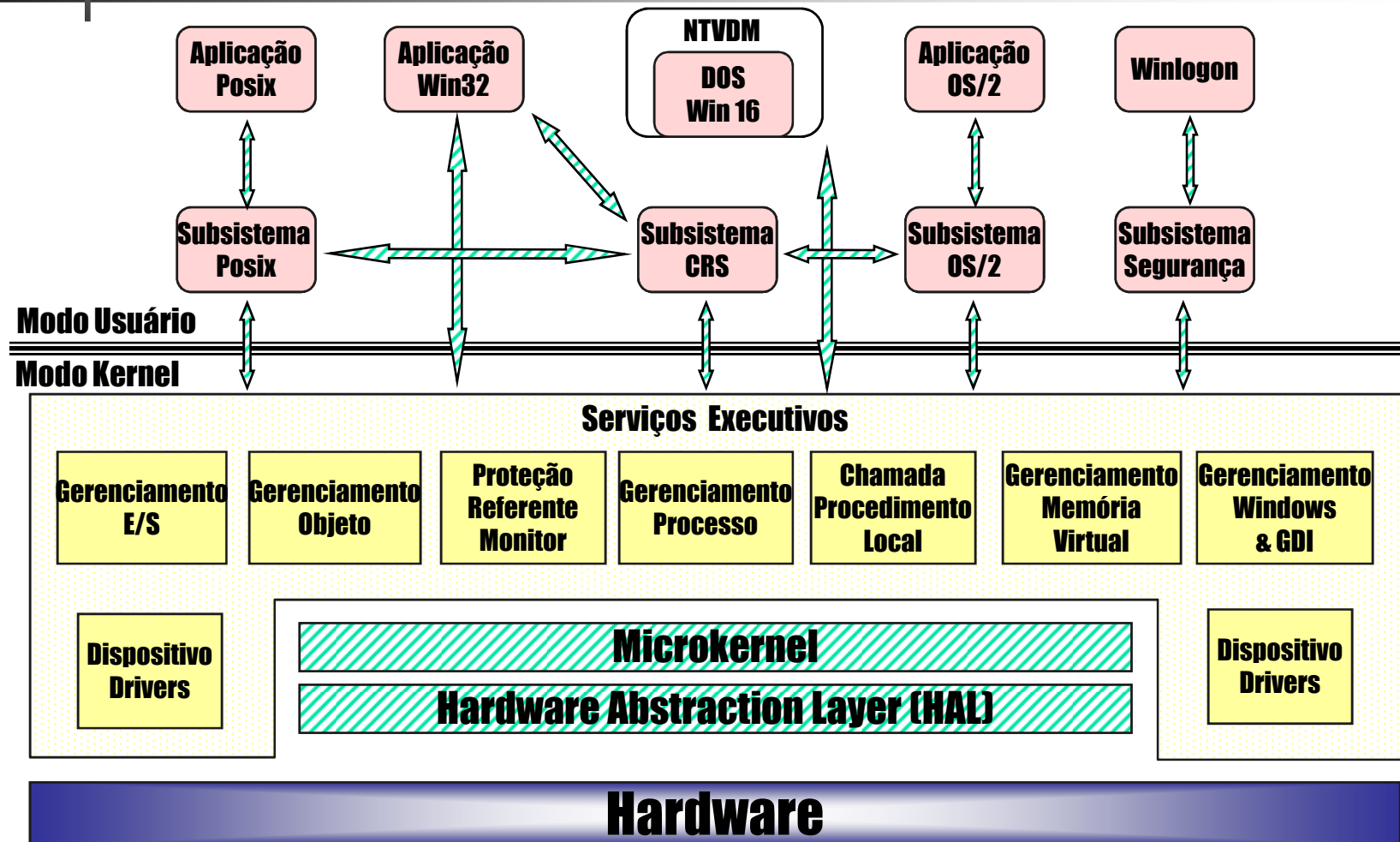


Estruturas do S.O. Arquitetura do Unix



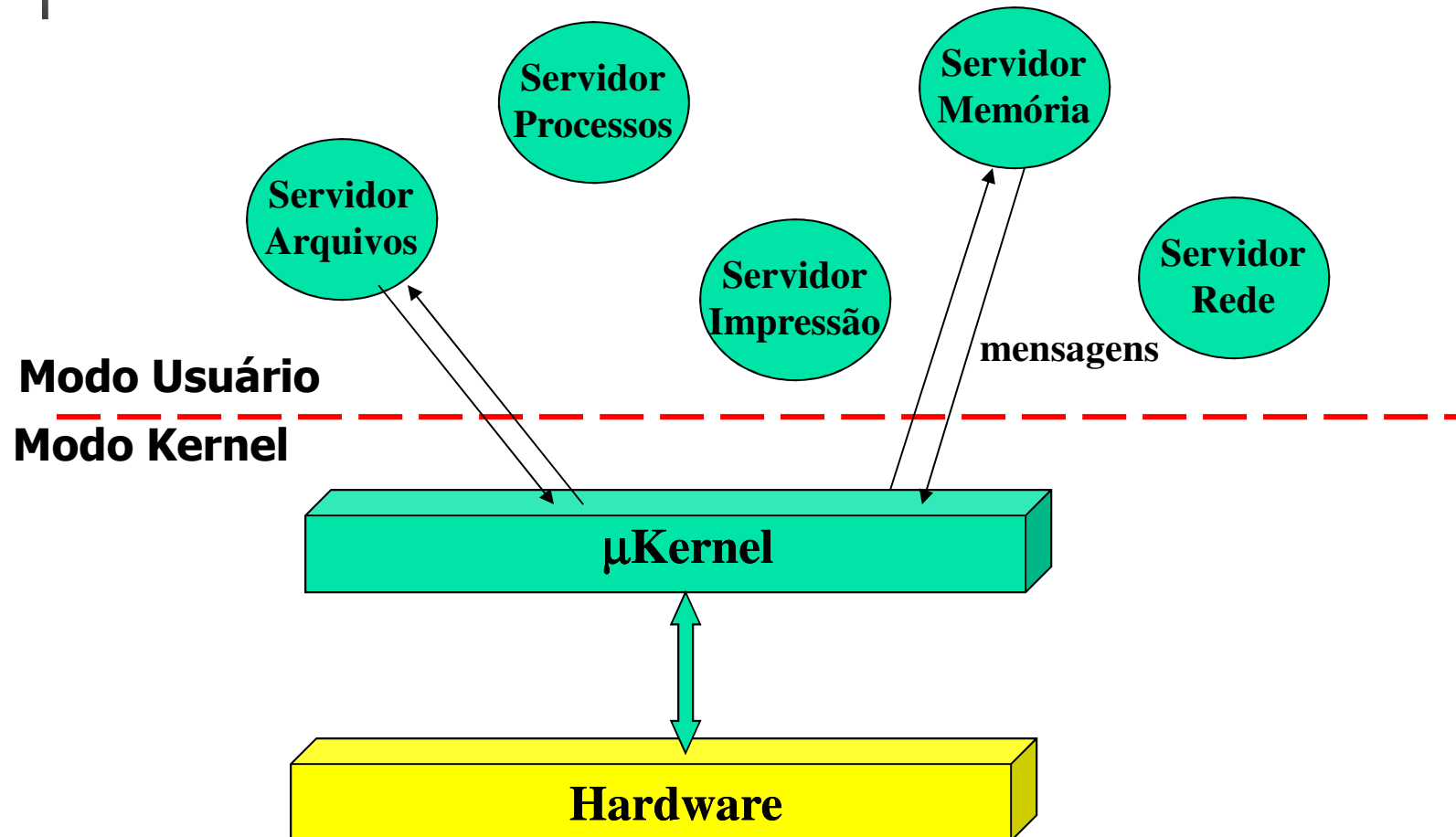


Estruturas do S.O. Arquitetura Windows NT



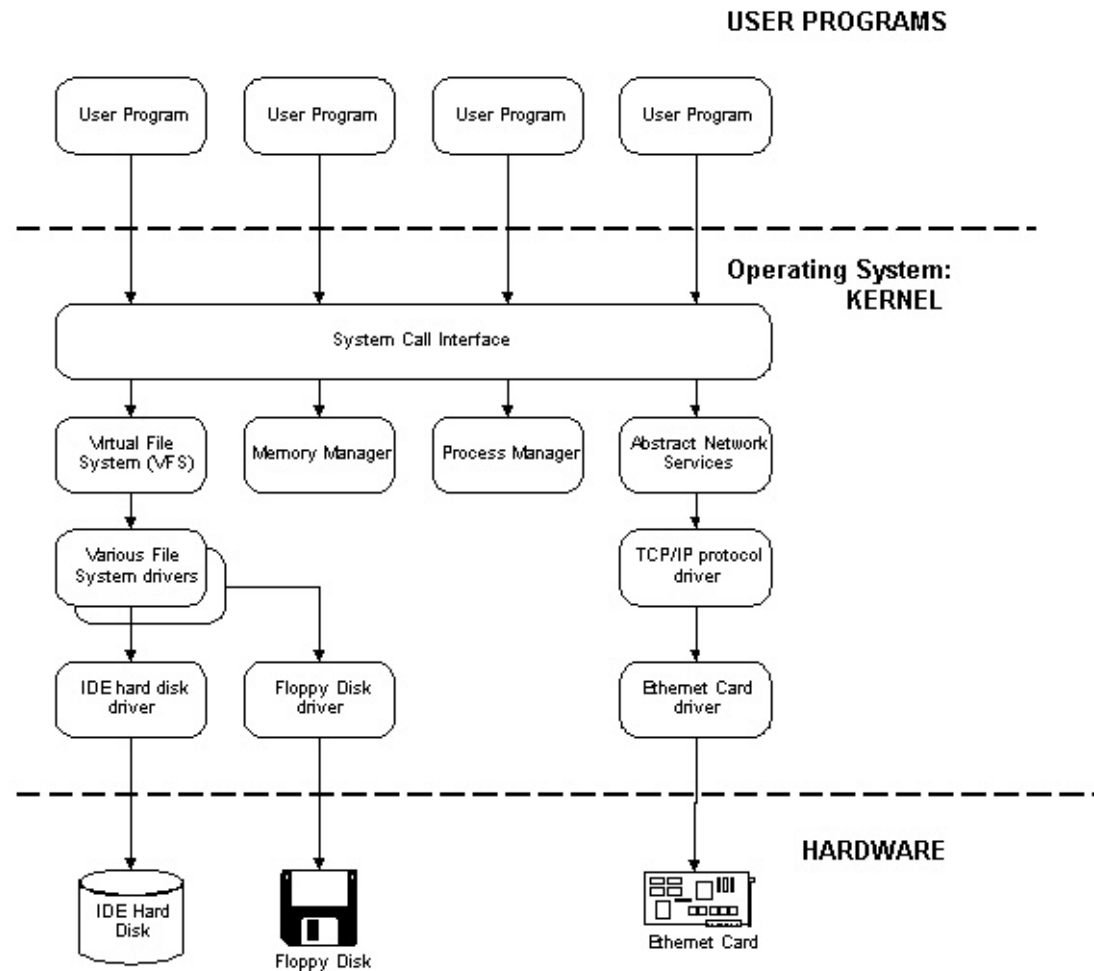


Estruturas do S.O. Arquitetura μ Kernel



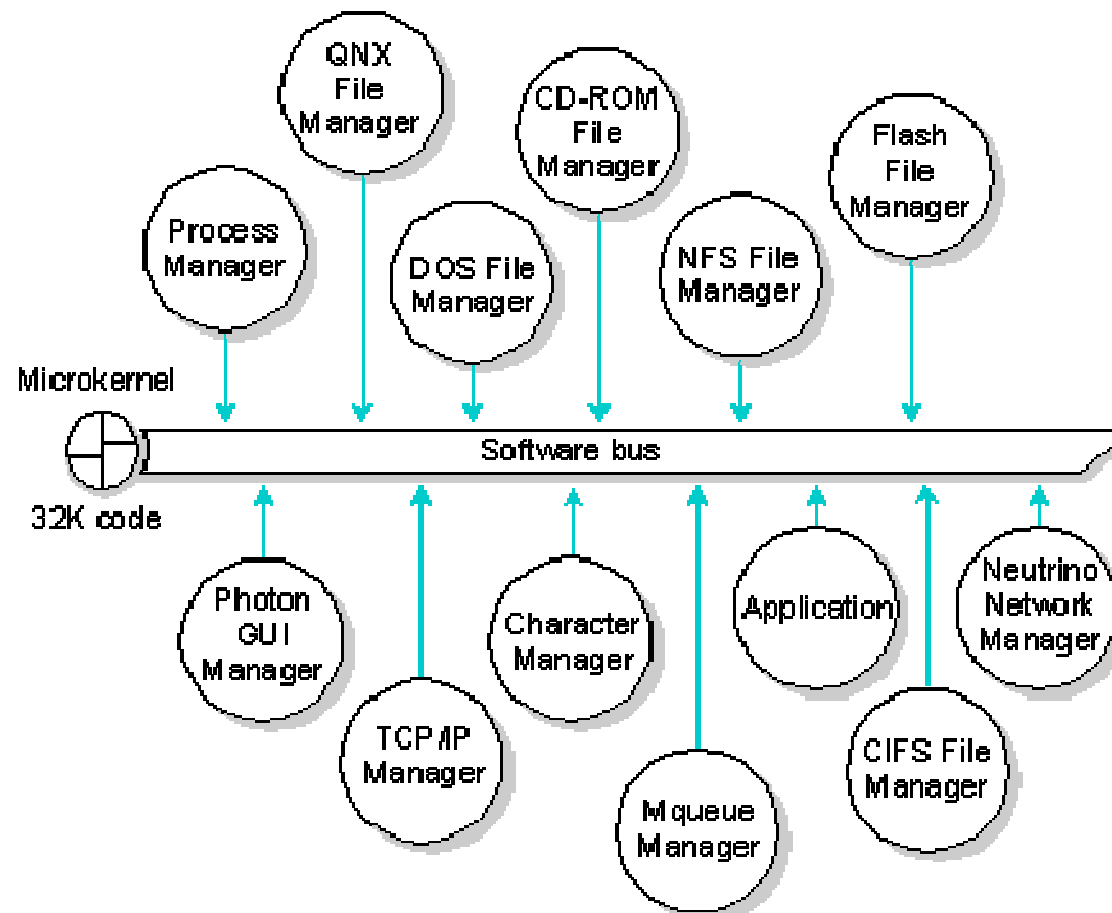


Estruturas do S.O. Arquitetura do Linux





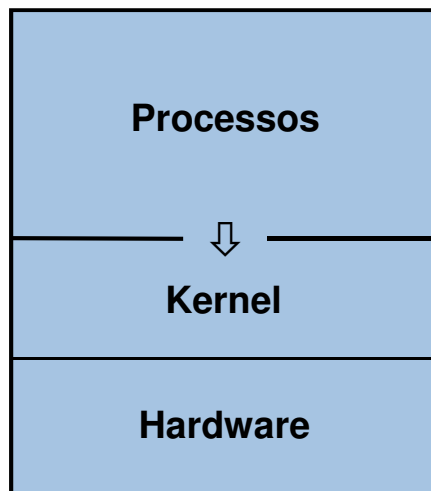
Estruturas do S.O. Arquitetura do QNX



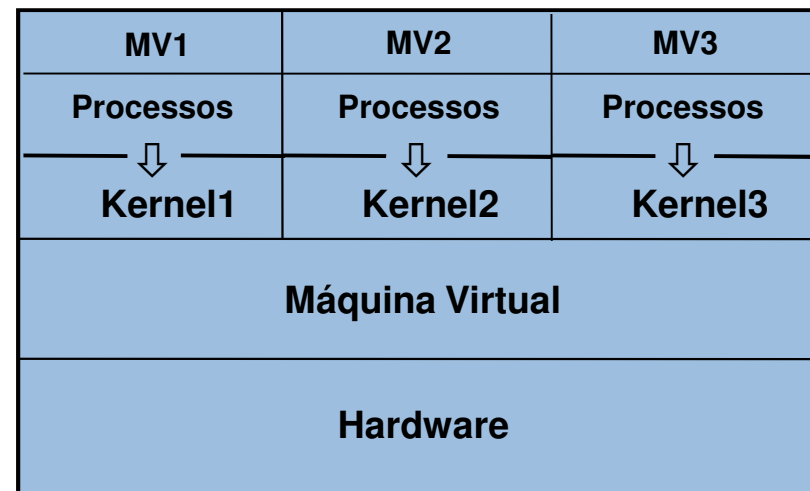


Estruturas do S.O. Máquina Virtual

Uma máquina real pode abrigar internamente diferentes ambientes virtuais, cada um simulando uma máquina distinta, com memória, SO, recursos e processos próprios. Desta forma, cada usuário (ou aplicativo) parece possuir sua própria máquina



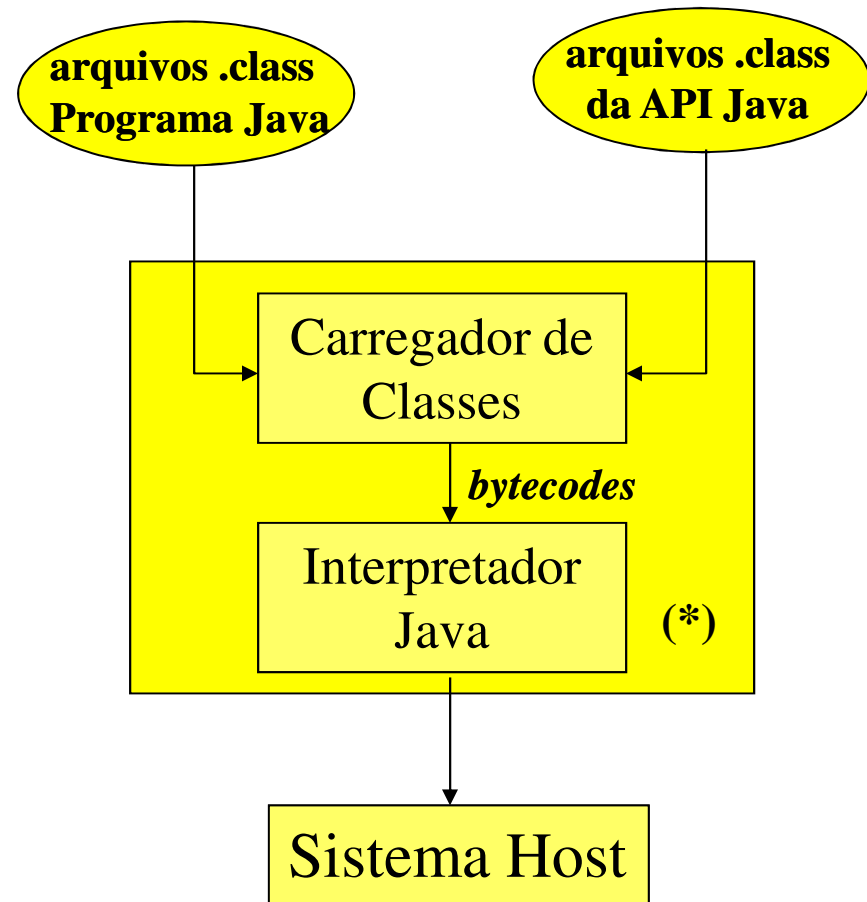
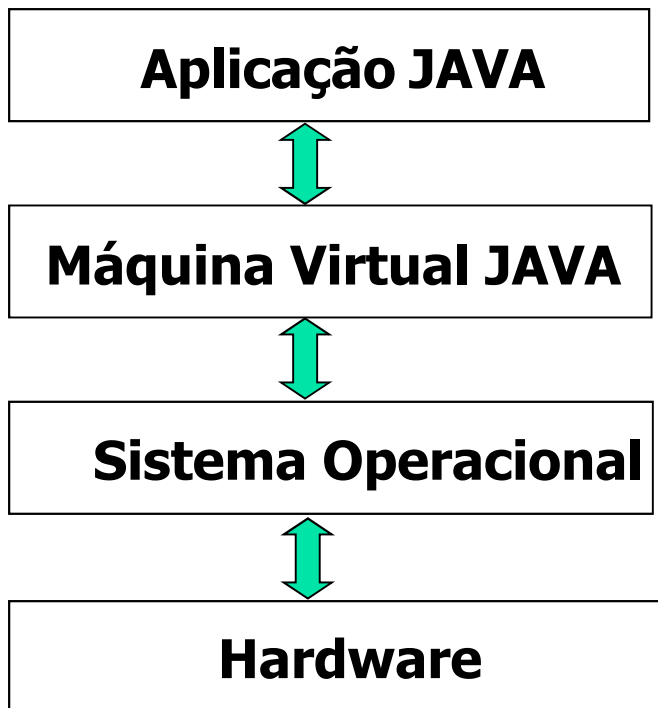
Máquina Não-Virtual



Máquina Virtual



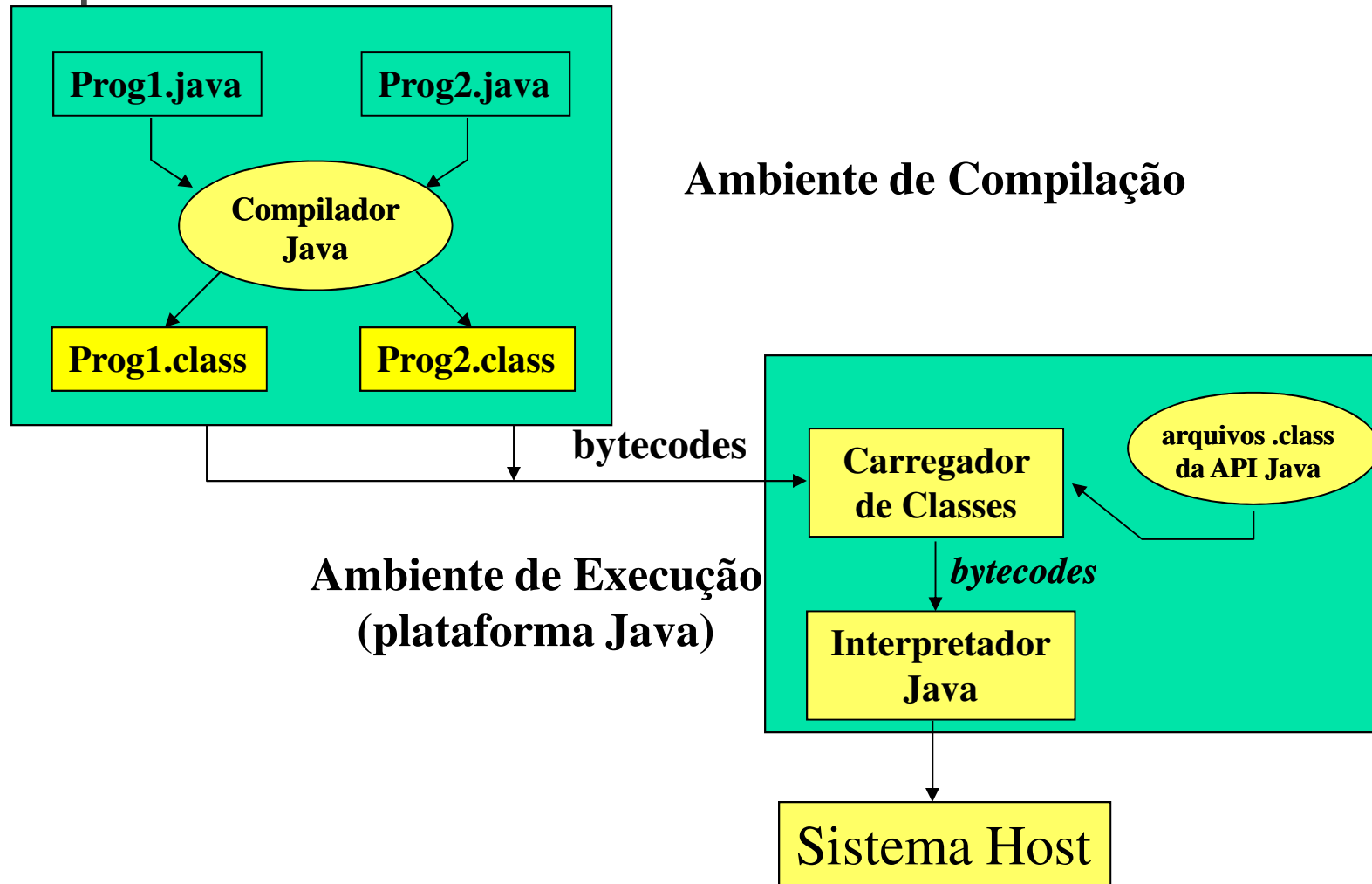
Estruturas do S.O. Máquina virtual Java



(*) Dependente do Hardware



Estruturas do S.O. Ambiente de desenvolvimento Java





- Execução de programas
- Operações de E / S
- Manipulação de Arquivos
- Comunicação entre processos
- Detecção de Erros
- Alocação de Recursos
- Contabilização
- Proteção

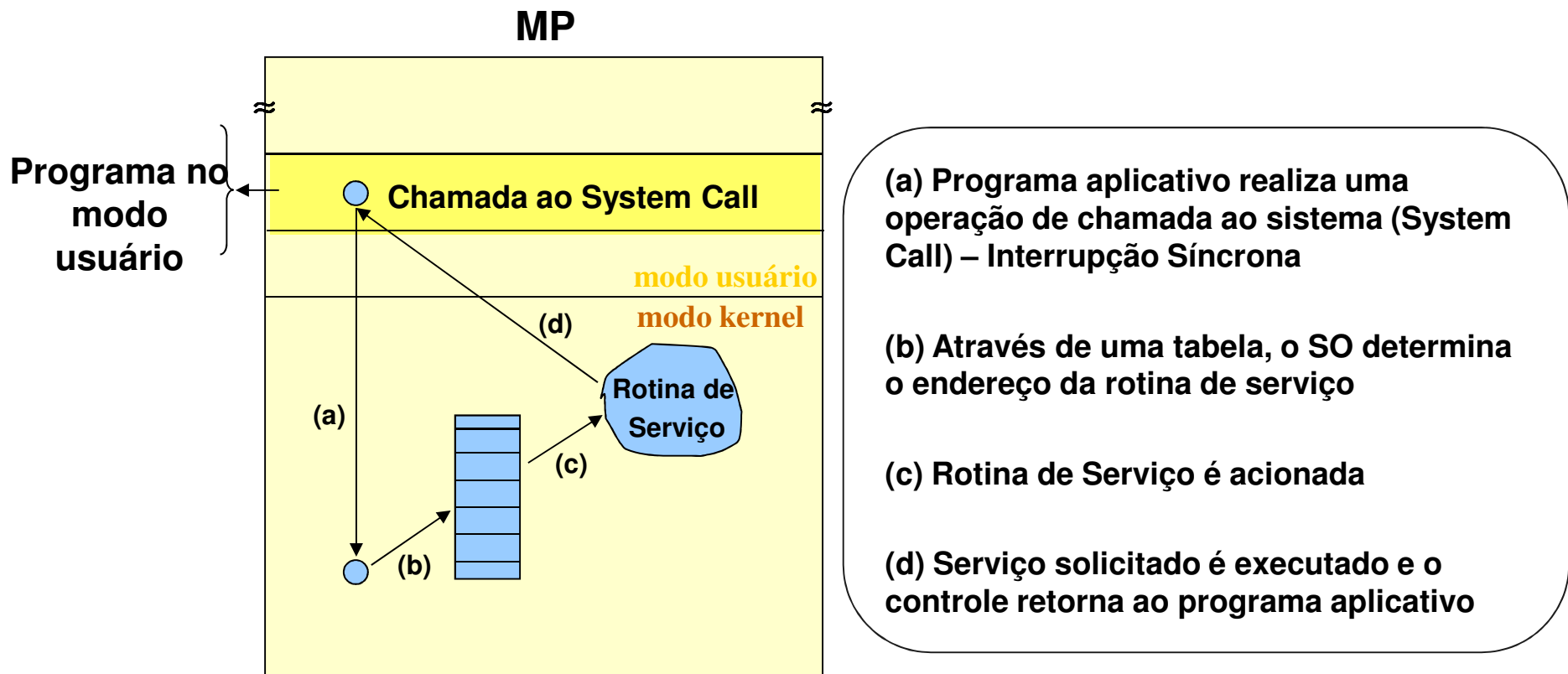
**Chamadas ao Sistema
(system calls)**



Estruturas do S.O.

Chamadas ao Sistema

Fornecem a interface entre o SO e os processos





- Suspendem a tarefa em execução pela ocorrência de um evento externo (interrupção)
- Permitem a execução de outras instruções enquanto uma operação de E/S está sendo executada
- Melhoram a eficiência do processador
- Acionam uma Rotina do SO chamada de Tratador de Interrupções – “Interrupt Handler”

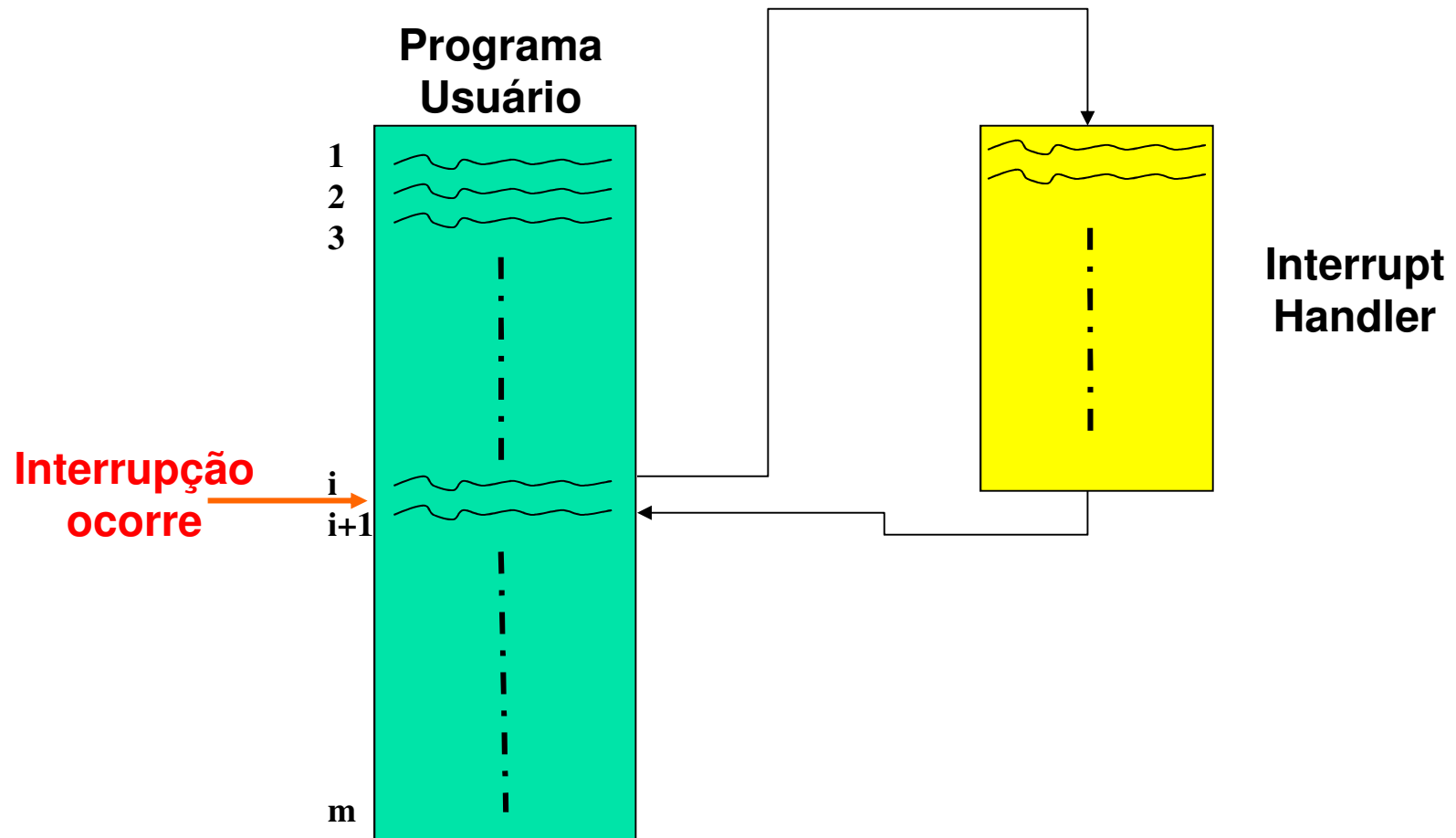
Observação:

Após o término da interrupção,
a tarefa suspensa pode retornar
à execução ou uma outra ser selecionada.



Estruturas do S.O.

Tratamento de Interrupções





Tratamento de interrupções

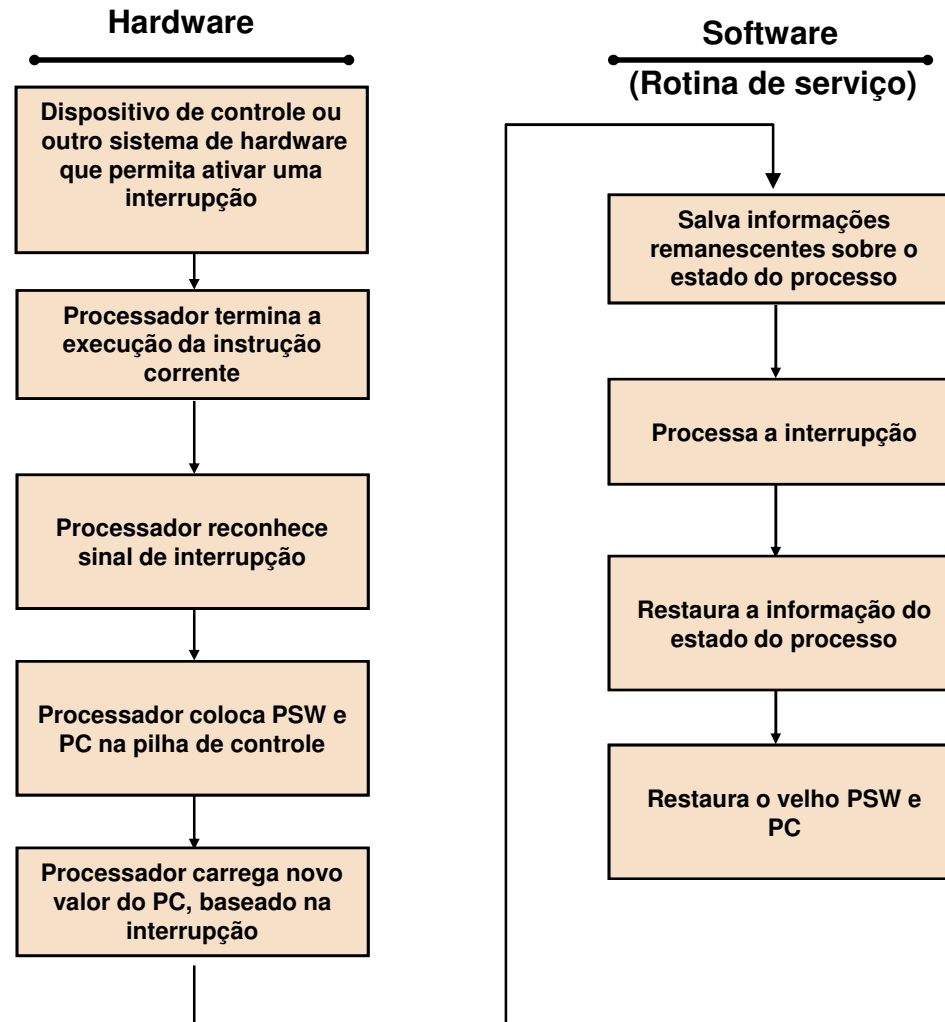
É feito pelo SO, que determina a natureza da interrupção e dispara a Rotina de Serviço adequada para executar as ações que forem necessárias.

- A execução do programa corrente é suspensa
- O endereço da Rotina de Serviço é localizado na tabela de interrupções
- O status do programa corrente é salvo (conteúdo do PC, PSW, registradores, ...)
- O controle do processador é transferido para a rotina de serviço
- O ponto de interrupção pode ocorrer em qualquer ponto do programa
- As rotinas de serviço normalmente fazem parte do Sistema Operacional
- Overhead adicional para ativar e executar a rotina de serviço.



Estruturas do S.O.

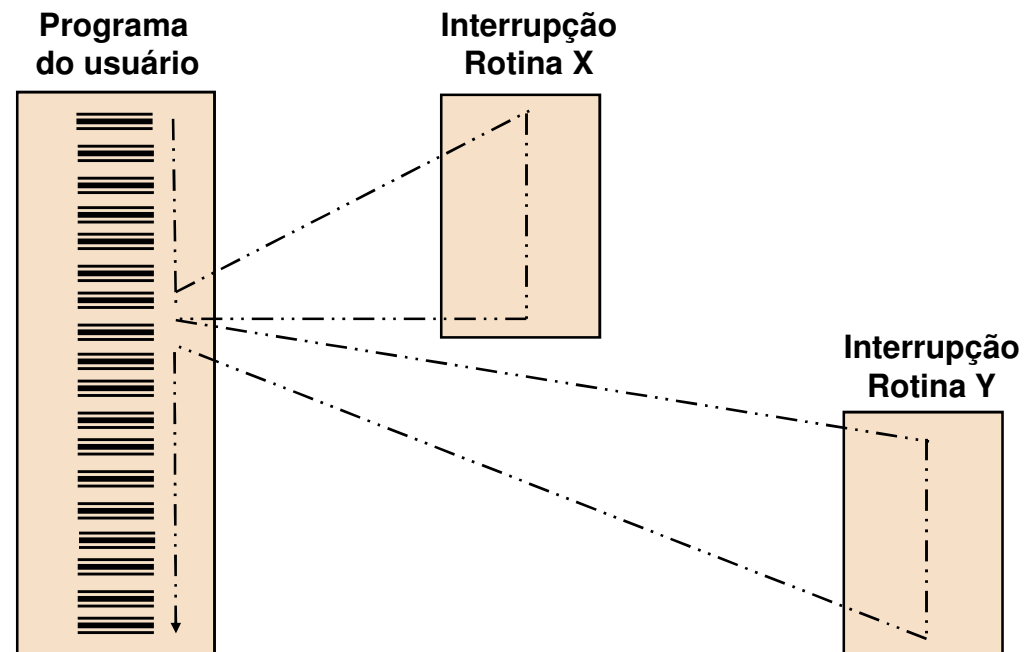
Fluxo de uma interrupção





Múltiplas Interrupções – Modelo Sequencial

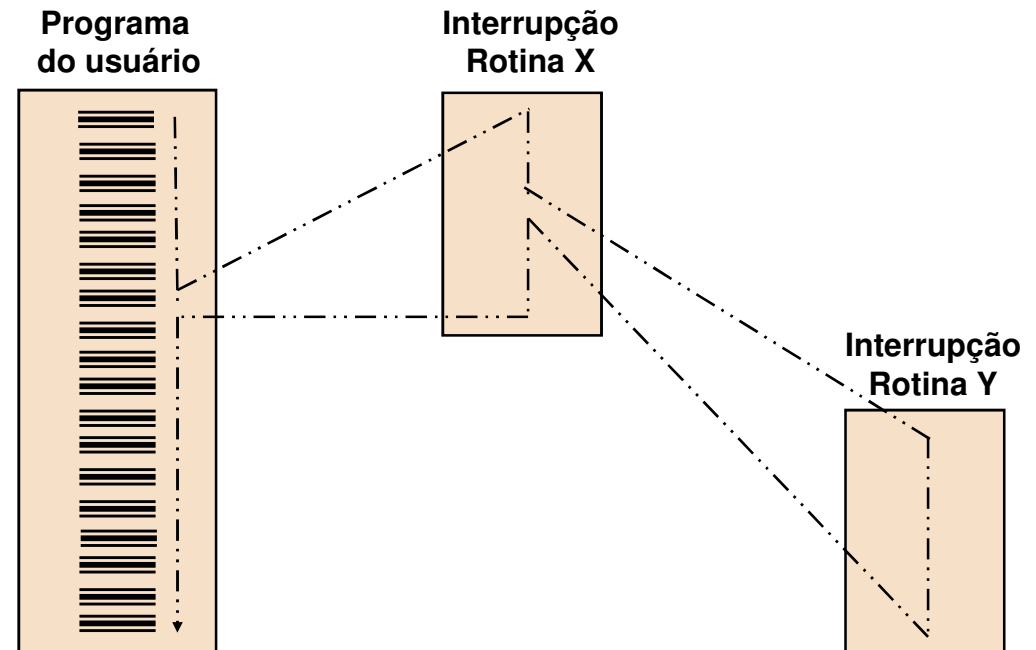
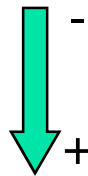
- A Rotina de Serviço desabilita as interrupções
- Uma nova interrupção só é tratada após o retorno
- A interrupção pode demorar a ser tratada, o que pode eventualmente ocasionar uma perda de dados
- Finalizada a Rotina de Serviço de interrupção, o processador checa por interrupções adicionais





Múltiplas interrupções – modelo cascata

- Interrupções têm prioridade
- Interrupções com alta prioridade interrompem rotinas de serviço de interrupções de menor prioridade
- Exemplos de prioridade:
 - impressora
 - disco
 - comunicação





Estruturas do S.O.

Tipos de Interrupções

Síncrona

Estados de Exceção (trap)

- ✓ estouro aritmético
- ✓ divisão por zero
- ✓ instrução ilegal
- ✓ acesso não permitido

Interrupção de software

- ✓ chamada de sistema (system call)

Relógio (temporizador)

- ✓ usado pelo programa
- ✓ usado pelo SO (time slice)

Assíncrona

Falha de Hardware

- ✓ Erro de paridade Memória
- ✓ Falha no disco, etc

Entrada e Saída

- ✓ Sinalização de conclusão



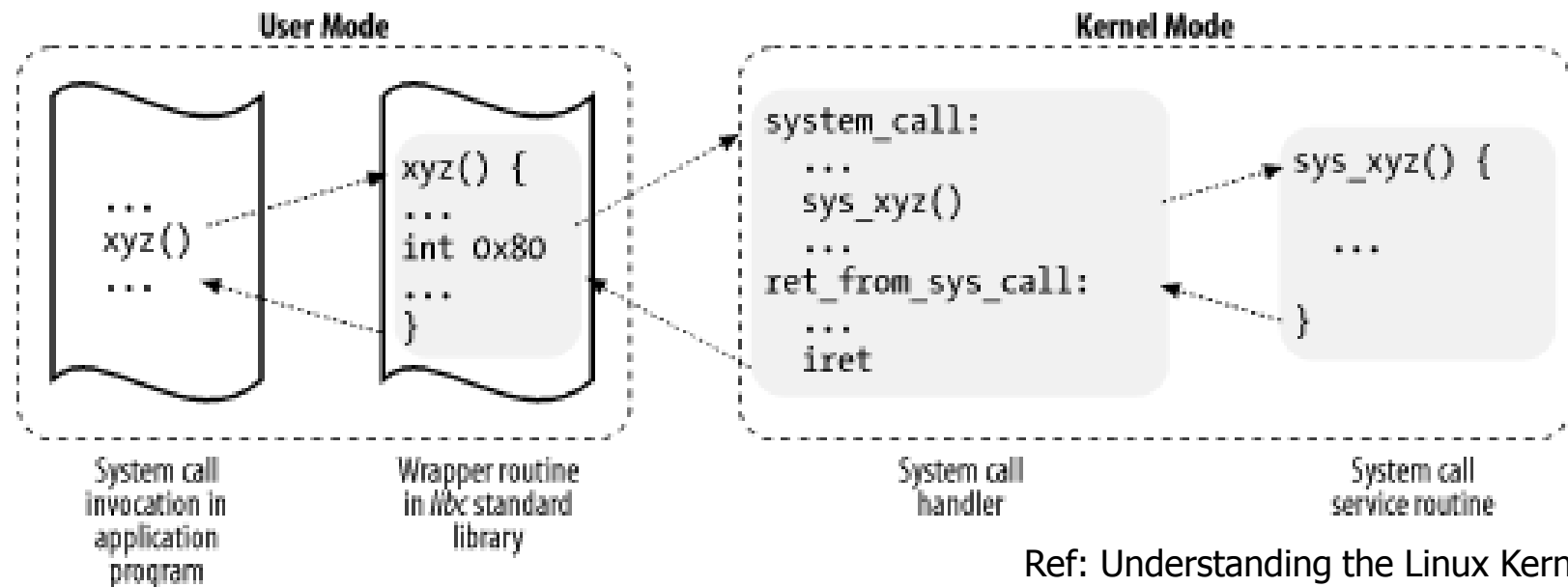
Estruturas do S.O. Exceções no 80386

- 0** Erro de Divisão (Divide Error)
- 1** Exceções de Debug (Debug Exceptions)
- 2** -----
- 3** Ponto de parada de execução (Breakpoint)
- 4** Overflow
- 5** Checagem de Limites (Bounds Check)
- 6** Código Operacional Inválido (Invalid Opcode)
- 7** Coprocessador não disponível (Coprocessor Not Available)
- 8** Falha Dupla (Double Fault)
- 9** Segmento de Coprocessador Ultrapassado (Coprocessor Segment Overrun)
- 10** TSS Inválida (Invalid TSS)
- 11** Segmento Não Presente (Segment Not Present)
- 12** Exceção da Pilha (Stack Exception)
- 13** Exceção de Proteção Geral - Falha Tripla (General Protection Exception - Triple Fault)
- 14** Falha de Página (Page Fault)
- 15** -----
- 16** Erro do Coprocessador (Coprocessor Error)



Estruturas do S.O.

Execução de uma System Call no Linux



Ex: Linux `int 80h` → invoca system call
a identificação da systema call é
passada através do registrador
`eax`



Execução de uma System Call no Linux

Invocação

```
settimer (ITIMER_VIRTUAL, &timer, NULL);
```

Wrapper

```
_syscall1(unsigned int,alarm,unsigned int,seconds)
```

Mudança de modo

Handler

```
_system_call()
```

Rotina de serviço

```
unsigned int alarm(unsigned int seconds)
```



Estruturas do S.O. System Call no Linux

A lista de chamadas ao sistema no espaço do usuário estão contidas na libc e definidas em `<sys/syscall.h>`

`_llseek`, `_newselect`, `_sysctl`, `accept`, `access`, `acct`, `adjtimex`, `afs_syscall`, `alarm`, `bdflush`, `bind`, `break`, `brk`, `cacheflush`, `capget`, `capset`, `chdir`, `chmod`, `chown`, `chown32`, `chroot`, `clone`, `close`, `connect`, `creat`, `create_module`, `delete_module`, `dup`, `dup2`, `execve`, `exit`, `fchdir`, `fchmod`, `fchown`, `fchown32`, `fcntl`, `fcntl64`, `fdatasync`, `flock`, `fork`, `fstat`, `fstat64`, `fstatfs`, `fsync`, `ftime`, `ftruncate`, `ftruncate64`, `get_kernel_syms`, `getcwd`, `getdents`, `getdents64`, `getegid`, `getegid32`, `geteuid`, `geteuid32`, `getgid`, `getgid32`, `getgroups`, `getgroups32`, `getitimer`, `getpagesize`, `getpeername`, `getpmsg`, `getpgid`, `getpgrp`, `getpid`, `getppid`, `getpriority`, `getresgid`, `getresgid32`, `getresuid`, `getresuid32`, `getrlimit`, `getrusage`, `getsid`, `getsockname`, `getsockopt`, `gettid`, `gettimeofday`, `getuid`, `getuid32`, `gtty`, `idle`, `init_module`, `ioctl`, `ioperm`, `iopl`, `ipc`, `kill`, `lchown`, `lchown32`, `link`, `listen`, `lock`, `lseek`, `lstat`, `lstat64`, `madvise`, `mincore`, `mkdir`, `mknod`, `mlock`, `mlockall`, `mmap`, `modify_ldt`, `mount`, `mprotect`, `mpx`, `mremap`, `msync`, `munlock`, `munlockall`, `munmap`, `nanosleep`, `nfsservctl`, `nice`, `oldfstat`, `oldlstat`, `oldolduname`, `oldstat`, `oldumount`, `olduname`, `open`, `pause`, `personality`, `phys`, `pipe`, `pivot_root`, `poll`, `prctl`, `pread`, `prof`, `profil`, `ptrace`, `putpmsg`, `pwrite`, `query_module`, `quotactl`, `read`, `readahead`, `readdir`, `readlink`, `readv`, `reboot`, `recv`, `recvfrom`, `recvmsg`, `rename`, `rmdir`, `rt_sigaction`, `rt_sigpending`, `rt_sigprocmask`, `rt_sigqueueinfo`, `rt_sigreturn`, `rt_sigsuspend`, `rt_sigtimedwait`, `sched_get_priority_max`, `sched_get_priority_min`, `sched_getparam`, `sched_getscheduler`, `sched_rr_get_interval`, `sched_setparam`, `sched_setscheduler`, `sched_yield`, `security`, `select`, `sendfile`, `send`, `sendmsg`, `sendto`, `setdomainname`, `setfsgid`, `setfsgid32`, `setfsuid`, `setfsuid32`, `setgid`, `setgid32`, `setgroups`, `setgroups32`, `sethostname`, `setitimer`, `setpgid`, `setpriority`, `setregid`, `setregid32`, `setresgid`, `setresgid32`, `setresuid`, `setresuid32`, `setreuid`, `setreuid32`, `setrlimit`, `setsid`, `setsockopt`, `settimeofday`, `setuid`, `setuid32`, `setup`, `sgetmask`, `shutdown`, `sigaction`, `sigaltstack`, `signal`, `sigpending`, `sigprocmask`, `sigreturn`, `sigsuspend`, `socket`, `socketcall`, `socketpair`, `ssetmask`, `stat`, `stat64`, `statfs`, `stime`, `stty`, `swapoff`, `swapon`, `symlink`, `sync`, `sysfs`, `sysinfo`, `syslog`, `time`, `times`, `truncate`, `truncate64`, `ulimit`, `umask`, `umount`, `uname`, `unlink`, `uselib`, `ustat`, `utime`, `vfork`, `vhangup`, `vm86`, `vm86old`, `wait4`, `waitpid`, `write`, `writev`.

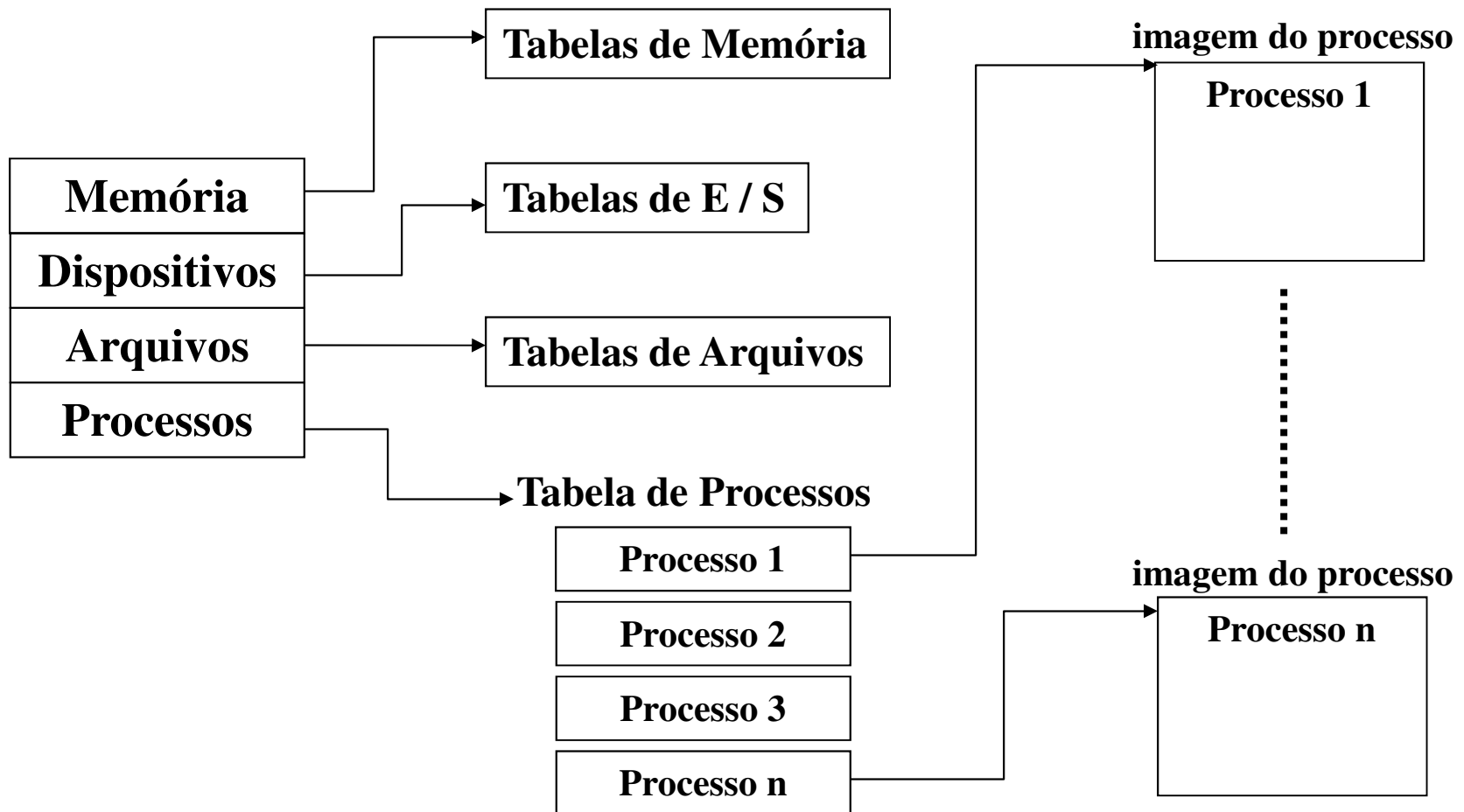


- Para navegar no código fonte com direito a ferramenta de busca use:

<http://lxr.linux.no>



Estruturas do S.O. Estruturas de controle





Estruturas do S.O. Sistemas Modernos - Tendências

- Arquitetura MicroKernel
- Sistema MultiThread
- Multiprocessamento Simétrico
- Distribuição
- Projeto Orientado a Objetos



Estruturas do S.O.

Tópicos a serem abordados

