
Estruturas de Sistemas Operacionais

Prof. Dr. José Luís Zem

Prof. Dr. Renato Kraide Soffner

Prof. Ms. Rossano Pablo Pinto



Faculdade de Tecnologia de Americana

Centro Paula Souza

Tópicos

- ❑ Componentes do Sistema
- ❑ Serviços do Sistema
- ❑ Chamadas do sistema
- ❑ Programas do Sistema (Utilitários)
- ❑ Estrutura do Sistema
- ❑ Projeto de Sistemas
- ❑ Implementação de Sistemas e Geração do Sistema

Componentes do Sistema

- ❑ Gerência de Processos
- ❑ Gerência de Memória Principal
- ❑ Gerência de Arquivos
- ❑ Gerência do Sistema de I/O
- ❑ Gerência de Armazenamento Secundário
- ❑ Rede
- ❑ Sistema de Proteção
- ❑ Interpretador de Comandos do Sistema

Gerência de Processos

- Processo é um programa em execução e necessita de certos recursos, como tempo de CPU, memória, arquivos e dispositivos de I/O, para completar sua tarefa.
- Responsabilidades do Sistema Operacional:
 - criar e destruir processos do usuário e do sistema;
 - suspender e reativar processos;
 - fornecer mecanismos para a comunicação entre processos;
 - fornecer mecanismos para a sincronização entre processos;
 - fornecer mecanismos para o tratamento de deadlocks (impasses).

Gerência de Memória Principal

- A memória principal é um armazenador de acesso rápido e compartilhável entre a CPU e os dispositivos de I/O, além de ser volátil, ou seja, perderá seu conteúdo em caso de falhas no sistema.
- Responsabilidades do Sistema Operacional:
 - conhecer quais partes da memória estão sendo usadas e por quem;
 - decidir quais processos serão carregados quando o espaço da memória torna-se disponível;
 - Alocar e liberar espaço da memória quando for necessário.

Gerência de Arquivos

- Um arquivo é uma coleção de informações definidas pelo seu criador. Normalmente representa programas (fonte, objeto e executável) ou dados.

- Responsabilidades do Sistema Operacional:
 - criar e remover arquivos;
 - criar e remover diretórios;
 - disponibilizar primitivas para manipulação de arquivos e diretórios;
 - mapear os arquivos no dispositivo de armazenamento secundário;
 - copiar arquivos entre dispositivos de armazenamento não voláteis.

Gerência do Sistema de I/O

- Ocultar as peculiaridades de dispositivos de *hardware* específicos do usuário.
- O subsistema de I/O consiste em:
 - *buffering, caching e spooling*;
 - interface geral de *driver* de dispositivo (*device-driver*);
 - *drivers* para dispositivos de hardware específicos.
- Apenas o *driver* de dispositivo conhece as peculiaridades do dispositivo específico ao qual foi atribuído.

Gerência de Armazenamento Secundário

- A memória principal (armazenamento primário) é volátil e muito pequena para armazenar todos os dados e programas permanentemente, portanto, o sistema computacional deve fornecer armazenamento secundário para auxiliar a memória principal.

- Responsabilidades do Sistema Operacional:
 - gerenciamento de espaço livre;
 - alocação de armazenamento;
 - escalonamento de disco (ordenação e seleção das operações p/ o uso de discos).

Rede

- Coleção de computadores compartilhando recursos.
- Cada computador tem seu próprio processador, memória e *clock*, além do sistema operacional.
- Os computadores são interligados através de um sub-sistema de comunicação e utilizam protocolos para viabilizar a comunicação.
- Acesso aos recursos compartilhados permitem:
 - *speed-up* computacional (aumento na velocidade de processamento);
 - aumentar a disponibilidade dos dados;
 - aumentar a confiabilidade.

Sistemas de Proteção

- Proteção refere-se a qualquer mecanismo para controlar o acesso de programas, processos ou usuários aos recursos do sistema ou de outros usuários.

- O mecanismo de proteção deve:
 - distinguir entre uso autorizado e não autorizado;
 - especificar o tipo de controle a ser utilizado;
 - fornecer os meios para viabilizar estes controles.

Interpretador de Comandos do Sistema

- ❑ Os comandos são enviados ao sistema operacional através de uma interface caracter ou orientada a mouse (“gráfica”).
- ❑ Comandos “ORIENTADOS A TECLADO” x “ORIENTADOS A MOUSE”
- ❑ Outros nomes: SHELL (casca do sistema), “prompt do sistema”, “console texto”, interpretador de linha de comando
- ❑ Os comandos interagem com:
 - o gerenciamento de processos;
 - manipulação de I/O;
 - gerenciamento de armazenamento secundário;
 - gerenciamento de memória principal;
 - proteção;
 - acesso à arquivos do sistema;
 - rede.

Serviços do Sistema Operacional

- Classes de serviço de um SO:
 - Execução de programas
 - Operações de I/O
 - Manipulação do Sistema de Arquivos
 - Comunicações
 - Detecção de Erros
 - Alocação de Recursos
 - Contabilização
 - Proteção

Serviços do Sistema Operacional

- Execução de programas:
 - capacidade do sistema para: carregar um programa na memória e executá-lo; terminá-lo normal ou anormalmente.

- Operações de I/O:
 - uma vez que os programas não podem executar operações de I/O diretamente, o sistema operacional deve fornecer algum meio para estas execuções.

- Manipulação do Sistema de Arquivos:
 - os programas necessitam ler, gravar, criar e remover arquivos.

Serviços do Sistema Operacional

- Comunicações
 - troca de informações entre processos que estão sendo executados em um mesmo computador ou em computadores diferentes interligados por uma rede. Implementado via memória compartilhada ou troca(envio) de mensagem.

- Detecção de Erros
 - garantir a correta computação detectando-se erros na CPU, memória, rede, dispositivos de I/O ou em programas do usuário.

Serviços do Sistema Operacional

- *Alocação de Recursos*
 - alocar recursos para múltiplos usuários ou múltiplos processos executando concorrente ou paralelamente (execução concorrente X execução paralela).

- *Contabilização*
 - registrar os usuários que utilizam os recursos do computador, quais tipos de recursos, em que quantidade e como os utilizam.

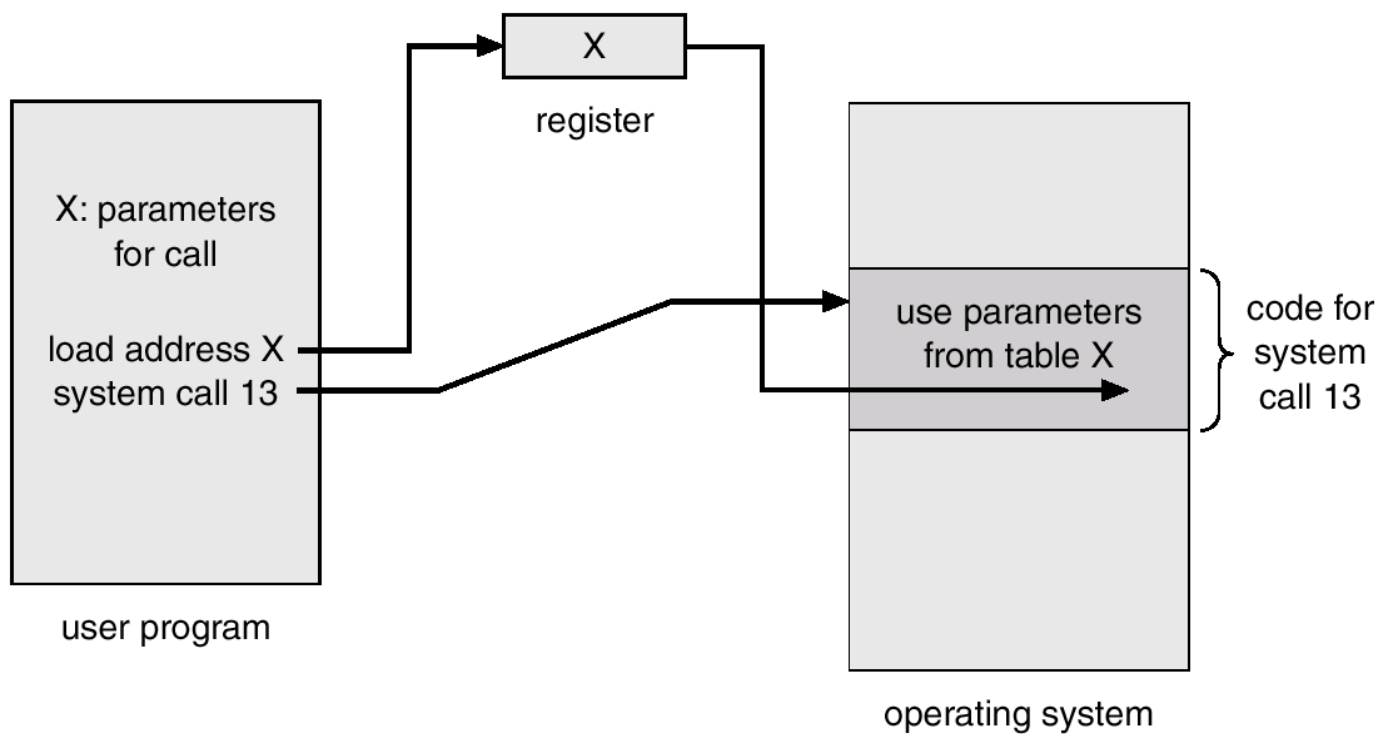
- *Proteção*
 - garantir que todos os acessos aos recursos do sistema sejam controlados. *“A resistência de uma corrente é a mesma de seu elo mais fraco.”*

Chamadas ao Sistema

- Chamadas ao sistema (*system calls*) fornecem uma interface entre um programa que está em execução e o sistema operacional.
- Três métodos podem ser utilizados para a passagem de parâmetros entre a aplicação e o sistema operacional:
 - passagem de parâmetros por meio dos registradores;
 - armazenamento dos parâmetros em posições da memória;
 - parâmetros podem ser inseridos em uma pilha pelo programa e retirados dela pelo sistema operacional.

Chamadas do Sistema

Passagem de Parâmetro



Chamadas do Sistema

```
#cpuid.s Sample program to extract the processor Vendor ID
#Author: Rossano Pablo Pinto (Baseado no código de Richard Blum)
.section .data
output:
.ascii "The processor Vendor ID is 'xxxxxxxxxxx'\n"
.section .text
.globl _start
_start:
    nop
    mov $0, %eax
    cpuid
    movl $output, %edi
    movl %ebx, 28(%edi)
    movl %edx, 32(%edi)
    movl %ecx, 36(%edi)
    movl $4, %eax      # USAR SYSCALL 4 (WRITE) P/ IMPRIMIR NA TELA (OLHAR /usr/include/asm/unistd_32.h)
    movl $1, %ebx      # IMPRIMIR EM STDOUT (FD 1)
    movl $output, %ecx # ENDERECO INICIAL DO TEXTO A SER IMPRESSO
    movl $42, %edx     # COMPRIMENTO DO TEXTO A SER IMPRESSO
    int $0x80          # "CHAMA" SYSCALL DO LINUX

    movl $1, %eax      # USAR SYSCALL 1 (EXIT) P/ FINALIZAR PROGRAMA
    movl $0, %ebx      # SAIR COM ERROR CODE = 0
    int $0x80          # "CHAMA" SYSCALL DO LINUX (INTERRUPÇÃO DE SOFTWARE)
```

Chamadas do Sistema

```
#ifndef _ASM_I386_UNISTD_H_
#define _ASM_I386_UNISTD_H_
```

```
/*
 * This file contains the system call numbers.
 */
```

```
#define __NR_restart_syscall    0
#define __NR_exit                1
#define __NR_fork                2
#define __NR_read                3
#define __NR_write               4
#define __NR_open                5
#define __NR_close               6
#define __NR_waitpid             7
#define __NR_creat               8
#define __NR_link                9
#define __NR_unlink              10
#define __NR_execve              11
#define __NR_chdir               12
#define __NR_time                13
#define __NR_mknod               14
#define __NR_chmod               15
#define __NR_lchown              16
#define __NR_break               17
#define __NR_oldstat             18
#define __NR_lseek               19
#define __NR_getpid              20
#define __NR_mount               21
#define __NR_umount              22
#define __NR_setuid              23
#define __NR_getuid              24
#define __NR_stime                25
```

```
#define __NR_ptrace                26
#define __NR_alarm                27
#define __NR_oldfstat             28
#define __NR_pause                29
#define __NR_utime                30
#define __NR_stty                 31
#define __NR_gtty                 32
#define __NR_access               33
#define __NR_nice                 34
#define __NR_ftime                35
#define __NR_sync                 36
#define __NR_kill                 37
#define __NR_rename               38
#define __NR_mkdir                39
#define __NR_rmdir                40
#define __NR_dup                  41
#define __NR_pipe                 42
#define __NR_times                43
#define __NR_prof                 44
#define __NR_brk                  45
#define __NR_setgid               46
#define __NR_getgid               47
#define __NR_signal               48
#define __NR_geteuid              49
#define __NR_getegid              50
#define __NR_acct                 51
#define __NR_umount2              52
#define __NR_lock                 53
#define __NR_ioctl                54
#define __NR_fcntl                55
#define __NR_mpx                  56
#define __NR_setpgid              57
#define __NR_ulimit                58
#define __NR_oldolduname          59
.....
```

Chamadas ao Sistema

- 5 grupos principais:
 - Controle de Processos
 - Gerência de Arquivos
 - Gerência de Dispositivos
 - Manutenção de Informações
 - Comunicações

Chamadas ao Sistema

- Controle de Processos
 - *end, abort (gera dump e controle é transferido p/ shell - retorna código de erro - 0 == normal)*
 - *load, execute (fork + exec)*
 - *create process (fork), terminate process (exit/signal)*
 - *get process attributes, set process attributes*
 - *wait for time, event*
 - *signal event (signal)*
 - *allocate and free memory*
- Gerência de Arquivos
 - *create and delete file*
 - *open and close file*
 - *read, write, reposition (seek)*
 - *get and set file attributes*

Chamadas ao Sistema

- Gerência de Dispositivos
 - *request and release device*
 - *read, write, reposition*
 - *get and set device attributes*
 - *logically attach and detach device*

- Manutenção de Informações
 - *get and set time or date, get and set system data*
 - *get and set process, file or device attributes*

- Comunicações
 - *create and delete communication connection*
 - *send and receive messages*
 - *transfer status information*
 - *attach and detach remote devices*

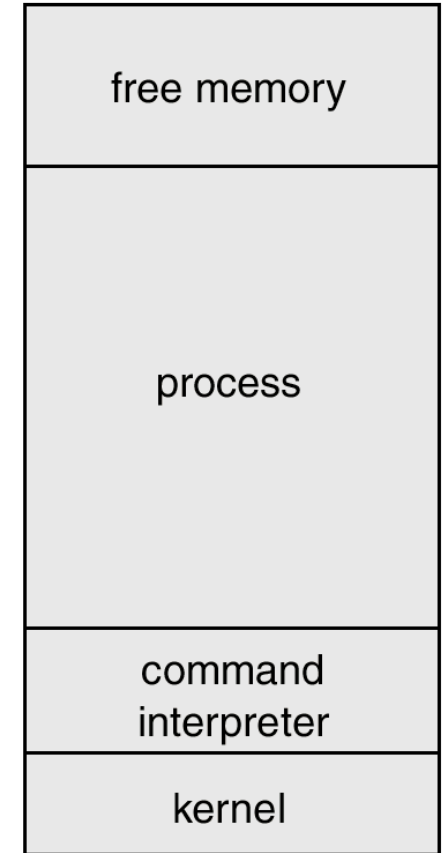
Execução no MS-DOS

Apenas 1 programa por vez:

- 1 - O programa é copiado p/ a memória
- 2 - O loader altera o PC do processador p/ apontar p/ a primeira instrução do programa recém carregado
- 3 - Ao final, o programa executa a chamada de finalização do programa
- 4 - O controle passa p/ o shell
- 5 - A memória ocupada pelo programa é liberada



(a)



(b)

Execução no MS-DOS

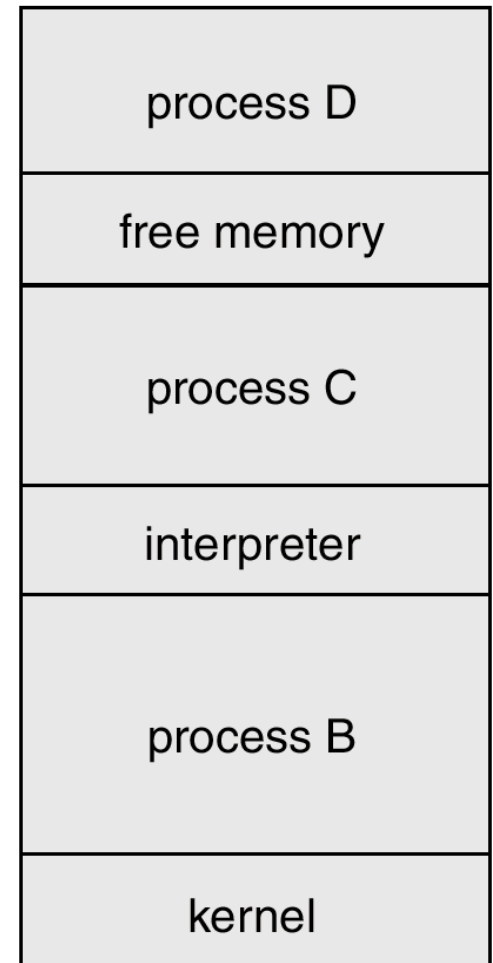
- O MS-DOS não suporta multiprogramação, mas suporta um método limitado de execução concorrente:
 - TSR (Terminate and Stay Resident) - um programa se conecta a uma interrupção e chama o serviço TSR

Execução no UNIX/Linux

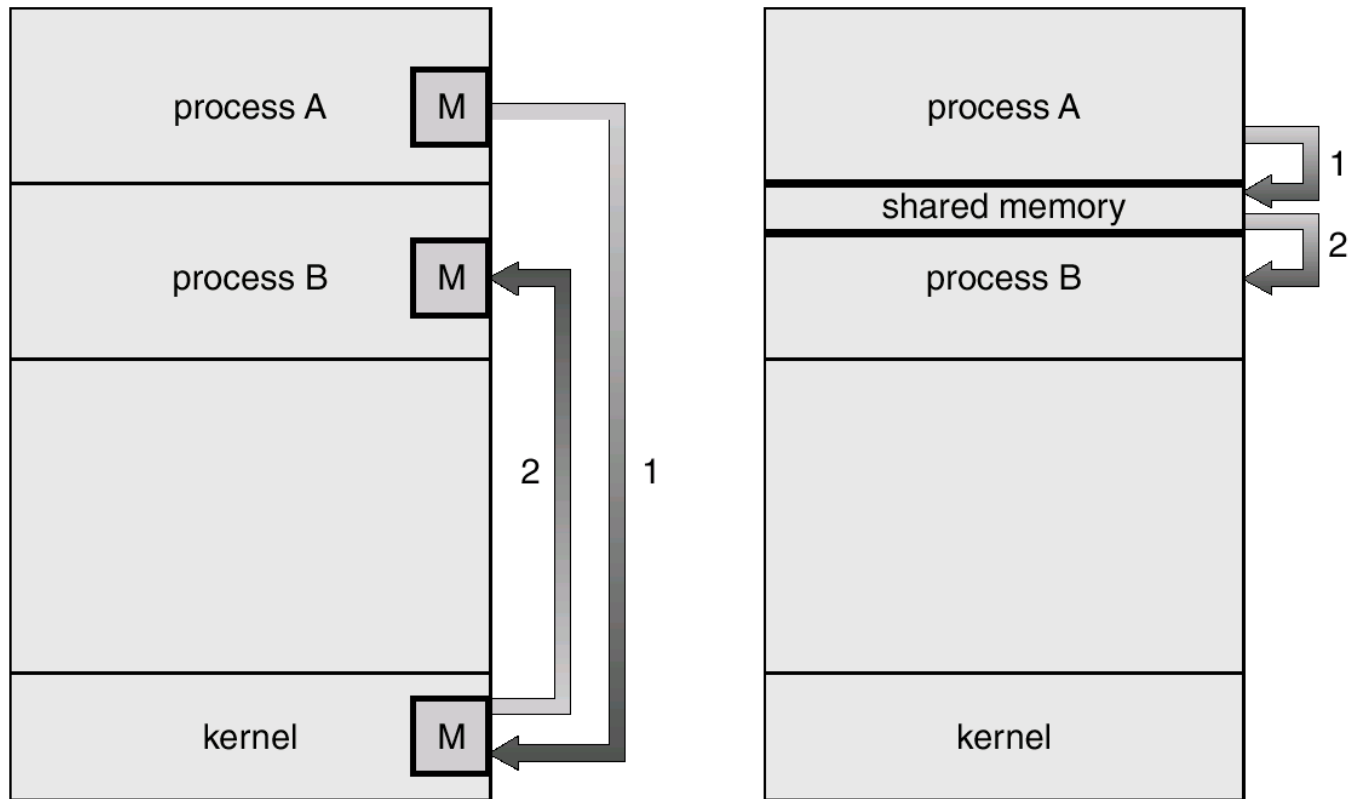
UNIX/Linux suportam multitarefa preemptiva.

Cada novo proceso no sistema é criado pela chamada `fork()` – feita pelo próprio interpretador de comandos (shell). O shell pode tanto esperar pelo término do processo quanto continuar a execução concorrente (background) - *nesse caso o processo em execução não pode obter dados do teclado, pois está sendo usado pelo shell.*

Um novo programa é carregado pela chamada `exec()`.



Modelos de Comunicação



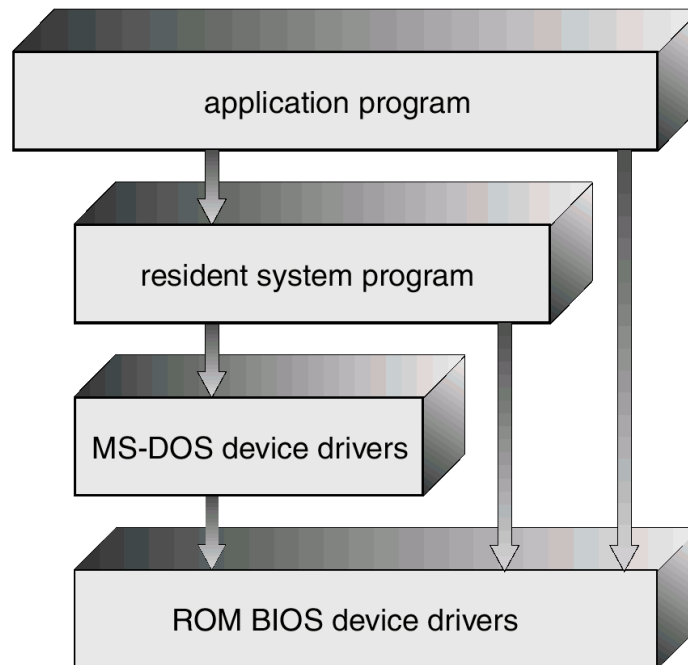
Programas do Sistema (Utilitários)

- Os programas do sistema fornecem um ambiente conveniente para o desenvolvimento e execução de programas. Eles podem ser divididos em:
 - Gerência de Arquivos
 - Informações de Status
 - Modificação de Arquivo
 - Suporte a Linguagem de Programação
 - Carregamento e Execução de Programas
 - Comunicações

- A **visão** que o **usuário** tem do sistema operacional é definida pelos programas do sistema e não pelas chamadas do sistema (**visão** do **programador**).

Estrutura do Sistema MS-DOS

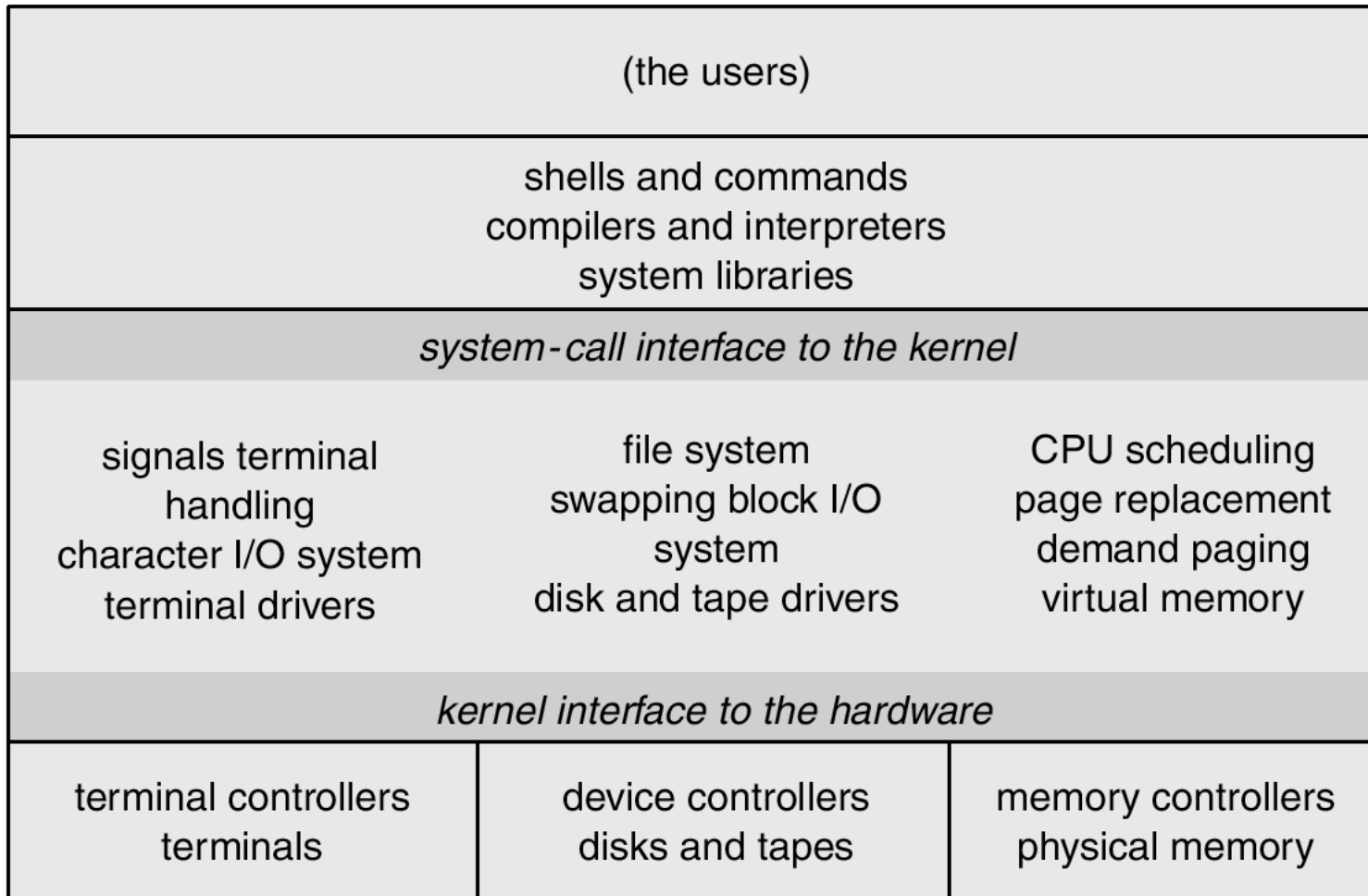
- O MS-DOS foi escrito para fornecer uma maior funcionalidade ocupando o menor espaço possível.



Estrutura do Sistema Unix/Linux

- O UNIX/Linux consiste de duas partes separadas:
 - os programas do sistema;
 - o *kernel*
 - consiste em tudo que estiver abaixo da interface do sistema e acima do *hardware* físico;
 - fornece o sistema de arquivos, escalonamento da CPU, gerenciamento da memória e outras funcionalidades do sistema operacional.

Estrutura do Sistema Unix/Linux



Micro-Kernel

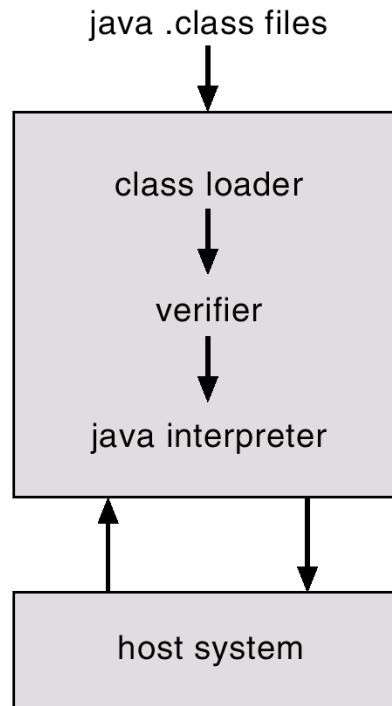
- Esta abordagem remove todos os componentes não-essenciais do kernel e implementa-os como programas do sistema e de nível do usuário.

- A comunicação entre os módulos é feita utilizando-se a passagem de mensagem.

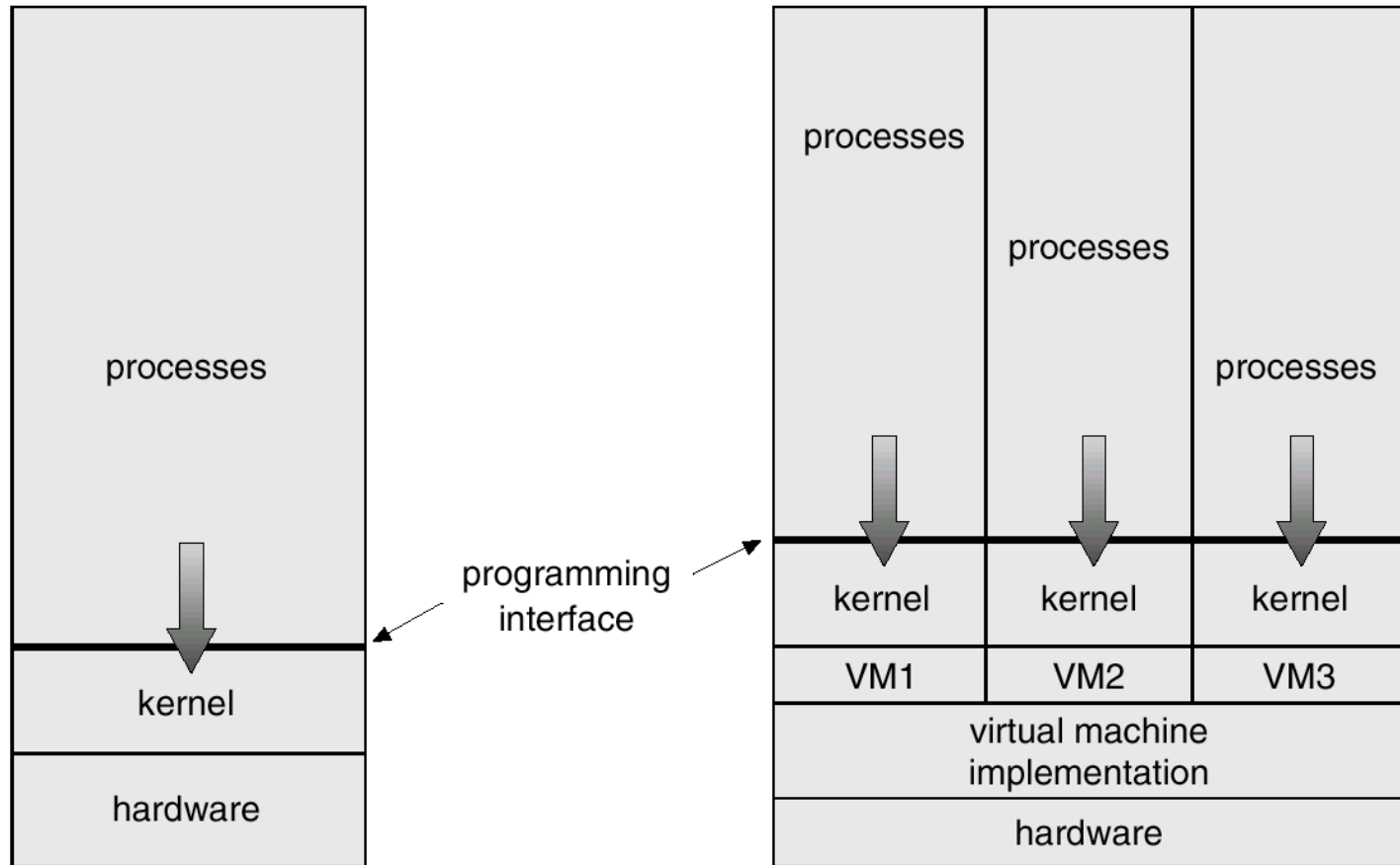
- Benefícios:
 - expansão do micro-kernel;
 - facilidade de portabilidade;
 - maior confiabilidade (menor qte. de código);
 - maior segurança.

Máquina Virtual

- O sistema operacional cria a ilusão de múltiplos processos, cada um executando em seu próprio processador com sua própria memória e seu próprio sistema operacional.



Máquina Virtual (Virtualização)



Objetivos, Mecanismos e Políticas

- **Objetivos do Usuário:**
 - o sistema operacional deve ser conveniente para o uso, fácil de aprender, confiável, seguro e rápido.

- **Objetivos do Sistema:**
 - o sistema operacional deve ser fácil de projetar, implementar e manter, assim como ser flexível, confiável, livre de erros e eficiente.

- Os mecanismos determinam como tudo deverá se feito e as políticas determinam o que deverá ser feito.

- A separação das políticas dos mecanismos é um princípio importante. Ela permite a maximização da flexibilidade caso mudanças nas políticas venham a ocorrer.

Implementação do Sistema

- Os sistemas operacionais atuais são escritos utilizando-se linguagens de alto nível.
- Vantagens:
 - rapidez na escrita do código;
 - é mais compacta;
 - facilidade de desenvolvimento e debug;
 - portabilidade.

Configuração do Sistema

- Os sistemas operacionais podem ser gerados levando em consideração a plataforma destino.
 - Tipo de CPU
 - Quantidade de memória
 - Dispositivos presentes
 - Tipo de escalonamento/uso do sistema (Servidor/Desktop)
 - Etc..

- Ex.:
 - Linux foi portado para várias plataformas
 - O processo de configuração/compilação determina qual a plataforma que o sistema vai ser executado

Estruturas de Sistemas Operacionais

Prof. Dr. José Luís Zem

Prof. Dr. Renato Kraide Soffner

Prof. Ms. Rossano Pablo Pinto



Faculdade de Tecnologia de Americana

Centro Paula Souza