

# Sistemas Operativos

V.1.3 V.Lobo 2005

## INTRODUÇÃO

Sistemas Operativos

### • FUNÇÕES DO SISTEMA OPERATIVO

- Gestão de recursos
- Criação de uma máquina virtual



### • DIFERENÇAS ENTRE UMA MÁQUINA VIRTUAL E UMA MÁQ. REAL

- Input / Output
- Memória
- Sistema de ficheiros
- Protecção e tratamento de erros
- Interação de programas
- Controlo de programas

1

## História dos S.O.

Sistemas Operativos

- **Monitores de controlo**
  - Permitem o arranque do sistema, carregar programas, usar rotinas de I/O, eventualmente têm um interpretador.
- **Sistemas Batch**
  - Permitem carregar automaticamente o "JOB" seguinte, têm um controlo elementar da memória e dos processos.
- **Sistemas Multiprogramados**
  - Permitem vários processos concorrentes.
- **Sistemas Interactivos**
  - Permitem diálogo homem-máquina durante a execução.
- **Sistemas de Memória Virtual**
  - Gestão flexível da memória
- **Sistemas Distribuídos**
  - Gerem várias máquinas

2

## Tipos de Sistemas operativos

Sistemas Operativos

### • TIPOS DE SISTEMAS OPERATIVOS

- Mono-utilizador (por ex. MS-DOS)
- Controlo de processos (na industria, por ex. RTOS)
- Interrogação de ficheiros (base de dados)
- Processamento de transações (bancos, com. seg.)
- Uso geral (por ex. UNIX, VAX-VMS, OS/400)

### • SISTEMA DE USO GERAL

- Sistema operativo que tenta servir para tudo e a todos.
- Características:
  - Suporte de BATCH
  - CONCORRENTE
  - MULTI-ACESSO
  - INTERACTIVO

3

## Arquitectura

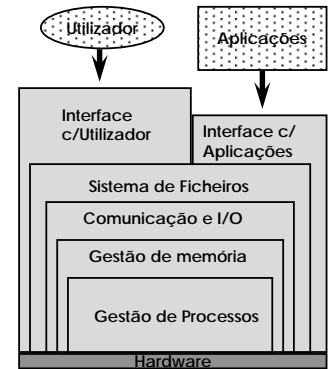
Sistemas Operativos

### • Para o utilizador

- É necessário um interpretador de comandos.

### • Para as aplicações

- É necessário uma biblioteca de funções do sistema operativo



4

## Principais problemas

Sistemas Operativos

### • Concorrência

- Decidir que programa deverá executar (SCHEDULING)
- Evitar "starvation" de processos
- Dar tempos de resposta aceitáveis
- Sincronizar os programas que têm necessidade disso
- Um S.O. é um sistema não determinístico !

### • Partilha de recursos

- Partilha de CPU, memória, periféricos
- Evitar que um programa interfira nos outros
- Decidir que informação ter em memória central, e qual deverá ser passada para disco

### • Armazenamento a longo prazo

- Sistema de ficheiros em disco

5

## CONCEITOS BÁSICOS

Sistemas Operativos

### • PROGRAMA

- Sequência de comandos sem actividade própria.

### • PROCESSO

- Em primeira aproximação é um programa a correr
- Pode-se chamar também tarefa
- Um processo pode envolver a execução de mais de um programa, inversamente, um determinado programa pode estar envolvido em mais de um processo
- É algo dinâmico, que existe num período limitado no tempo, enquanto um programa é algo estático que tem uma existência ilimitada

### • PROCESSADOR

- É um órgão material que executa uma acção definida numa instrução máquina.

6

# Sistemas Operativos

V.1.3 V.Lobo 2005

## CONCEITOS BÁSICOS

Sistemas Operativos

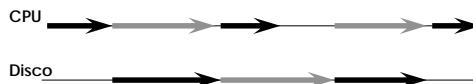
- **COMUNICAÇÃO ENTRE PROCESSOS**
  - Os processos no interior do sistema não actuam isoladamente, devem cooperar: troca de mensagens e memória partilhada
- **EXCLUSÃO MÚTUA**
  - Os recursos podem ser classificados em:
    - Partilháveis
    - Não partilháveis
  - Para os recursos não partilháveis, quando um processo o usa, é necessário excluir os outros
- **SINCRONIZAÇÃO**
  - A velocidade de um processo em relação a outro é imprevisível, visto que depende das interrupções e do tempo de processador que o sistema operativo atribui a cada processo. A certa altura é necessário que haja sincronização.

7

## GESTÃO DE PROCESSOS

Sistemas Operativos

- Atribuição de "time-slices" aos processos
- Existem vários critérios de atribuição de recursos
- Execução paralela é mais eficiente ( $1+1 < 2$ )
  - Cada processo pode demorar mais tempo
  - O desempenho global é melhor
  - Existe uma ocupação de recursos que nunca é perfeita

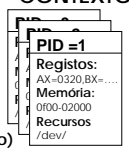


8

## GESTÃO DE PROCESSOS

Sistemas Operativos

- Métodos para interromper os processos
  - Métodos cooperativos
  - Métodos preemptivos
  - Importância de haver níveis de privilégios diferentes para o sistema operativo e para os processos
- Informação associada aos processos - CONTEXTO
  - Registos internos do processador
    - Importância de um stack próprio
  - Memória e recursos associados ao processo
  - Outras informações
    - Tempo de CPU gasto
    - Estado do processo (espera por um recurso)
  - Os processos são representados internamente por um DESCRITOR
- "Threads" ou processos "leves"

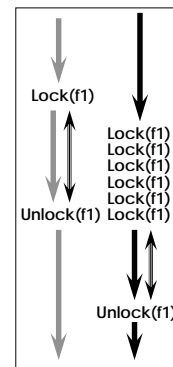


9

## SINCRONIZAÇÃO

Sistemas Operativos

- Sincronização é necessária para:
  - Exclusão mútua
  - Cooperação
  - Acesso a variáveis
  - Ferrolhos-Booleanos (ou flags)
- Sincronização com espera activa
  - while( not(flag1) )  
lock(flag1)
- Sincronização com espera passiva
  - O processo suspende a sua execução

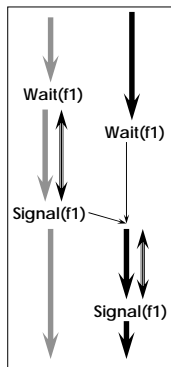


10

## SEMÁFOROS

Sistemas Operativos

- Servem para garantir exclusão mútua e sincronizar processos
  - Um processo só entra numa ZONA CRÍTICA se tiver "semáforo verde"
  - Ao entrar na ZONA CRÍTICA o processo levanta um "semáforo vermelho"
  - Os semáforos podem permitir mais do que um processo na ZONA CRÍTICA, ou seja podem ser "semáforos numerados"
  - Um semáforo é constituído por uma variável de controlo, uma função de entrada (*wait*), e uma função de saída (*signal*)
  - Em honra de Dijkstra, chama-se *p e v* a essas funções



11

## SEMÁFOROS

Sistemas Operativos

- **FUNÇÃO WAIT**
  - É a função de entrada numa zona crítica
  - Tem o aspecto *Wait(s)*, em que *s* é um semáforo.
  - Decrementa o valor de *s* se este for superior a zero. Se não, o processo fica bloqueado à espera que *s* torne o valor 1 ou superior.
- **FUNÇÃO SIGNAL**
  - Tem o aspecto *Signal(s)* em que *s* é um semáforo, isto é, uma variável inteira, positiva ou nula.
  - Esta função incrementa o valor de *s* (eventualmente desbloqueando um processo que esteja à espera)
- **DEADLOCK (INTER BLOCAGEM)**
  - Quando vários processos competem entre si no uso de recursos é possível que ocorra a situação em que nenhum processo possa continuar, porque os recursos que um precisa estão reservados pelo outro e vice-versa.

12

# Sistemas Operativos

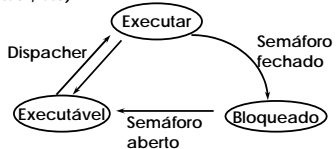
V.1.3 V.Lobo 2005

## DESPACHO

Sistemas Operativos

### • Estados possíveis de um processo

- Os processos são passados de um estado para outro por um programa dedicado, chamado DISPATCHER (ou despacho)
- Os processos podem estar a EXECUTAR no CPU, podem estar EXECUTÁVEIS, mas em lista de espera para o CPU, ou BLOQUEADOS em semáforos
- Num sistema mais completo, um processo pode ainda estar suspenso em disco.
- Cada processo é representado por um *descriptor* que contém toda a informação relevante (conteúdo dos registos do CPU, incluindo PC, SP, Acc, etc, nome do processo, prioridade, estado actual, etc)



13

## Gestão de memória

Sistemas Operativos

### • Hierarquia de memória

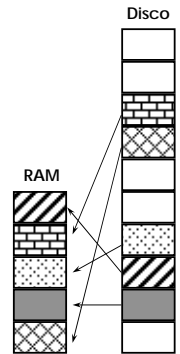
- Cache / Principal / Disco (ou secundária)

### • Endereçamento real / virtual

- O Sis.Op. por vezes simula os endereços de memória
- Torna a programação mais flexível

### • Memória virtual

- SEGMENTAÇÃO (prob.: fragmentação)
- PAGINAÇÃO
- Algoritmos de trocas de páginas:
  - LRU (least recently used)
  - MRU (most recently used)
  - LFU (least frequently used)
  - FIFO (first in, first out)



14

## Comunicação entre processos

Sistemas Operativos

### • Modelo de troca de mensagens

- Caixas de correio
- Canais virtuais
- Formatos de mensagens
- Invocação de rotinas remotas

### • Modelo de memória partilhada

- Variáveis comuns

### • Modelos de produtor/consumidor

- Mestre/escravo
- Correio
- Diálogo

15

## MS-DOS

Sistemas Operativos

### • Características

- Mono-processo (logo mono-utilizador)
- Desenhado para uma máquina apenas (IBM-PC e seus derivados)
- Interface com o utilizador em modo comando (não gráfico)

### • Estrutura

- BIOS
- Kernel
- Processador de comandos

### • BIOS

- Composto pelos DEVICE DRIVERS do bios.sys (ou io.sys) e outros que são instaláveis através do comando DEVICE= do config.sys
- Está sempre em memória
- Gere os devices CON, PRN, AUX, clock, etc.

16

## MS-DOS

Sistemas Operativos

### • Kernel

- Implementa as funções de sistema disponíveis para os outros programas (quase todas através da interrupção INT 21)
- É carregado em memória a partir do msdos.sys

### • Exemplo de uma chamada ao MS-DOS

```
mov AH, 3FH ; Código da função a chamar
mov BX, 0   ; um parametro
mov CX, 80  ; outro parametro (n.de bytes)
mov DS, segbuff ; outro parametro
mov DX, offsetbuff ; outro parametro
int 21H    ; chamada da função
```

17

## MS-DOS

Sistemas Operativos

### • Processador de comandos

- Faz a interface com o utilizador
- Implementado pelo COMMAND.COM
  - Parte residente
  - Secção de inicialização
  - Módulo removível

### • Parte residente

- Nunca sai de memória, permite re-carregar o command.com, trata erros como o ctrl-C, o "abort, retry, ignore?", etc

### • Secção de inicialização

- Carrega e interpreta o AUTOEXEC.BAT

### • Módulo removível

- contém a maior parte do interpretador, mas pode ser tirado de memória para que os outros programas tenham mais espaço

18

# Sistemas Operativos

V.1.3 V.Lobo 2005

## MS-DOS

Sistemas Operativos

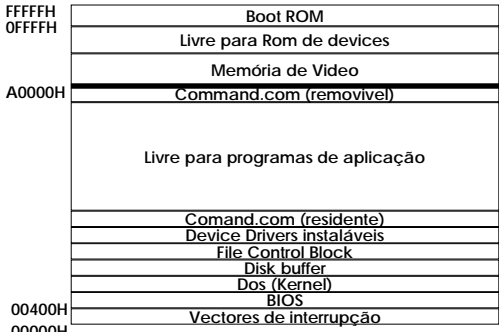
- **Categorias de comandos**
  - Comandos internos
    - contidos no `command.com`
  - Comandos externos
    - carregados como qualquer outro programa
  - O `command.com` tenta primeiro encontrar o programa dentro de si próprio (comando interno), procurando depois ficheiros com a extensão `.COM`, `.EXE`, e finalmente `.BAT`. Os ficheiros externos são procurados primeiro na directoria corrente, e depois em todas as que constam da variável de sistema `PATH`
- **Ficheiros .EXE e .COM**
  - contém código executável
  - os `.COM` usam apenas 1 página de memória (64K)
- **Ficheiros .BAT**
  - Ficheiros de texto com a invocação de comandos DOS

19

## MS-DOS

Sistemas Operativos

- **Mapa de memória**



FFFFFH  
0FFFFH

Boot ROM

Livre para Rom de dispositivos

Memória de Vídeo

A0000H

Command.com (removível)

Livre para programas de aplicação

Comand.com (residente)

Device Drivers instaláveis

File Control Block

Disk buffer

Dos (Kernel)

BIOS

00400H

00000H

Vectores de interrupção

20

## UNIX

Sistemas Operativos

- **HISTÓRIA**
  - Nasceu em 1969 do esforço isolado de cientistas dos laboratórios Bell, que tinham estado no projecto Multics
  - Início da década de 80: SystemV, BSD4, Xenix
  - Início da década de 90: aceitação geral, OSF
  - Para PCs: Xenix, Minix, e agora LINUX
  - Teve um forte impacto no meio universitário
  - Está muito ligado à linguagem C
  - Existem muitos "dialectos" para muitas máquinas
  - É um modelo para muitos outros sistemas operativos
  - Tem muitíssimos utilitários e programas de aplicação
- **Objectivos**
  - Gerir, em tempo partilhado, "pequenos" sistemas, de forma simples e eficiente

21

## Unix - Interface c/ Utilizador

Sistemas Operativos

- **Interface modo comando através de uma shell, que pode ser alterada.**
  - A shell por default é semelhante ao MS-DOS
  - Tipicamente os comandos têm uma série de *switches*
- **Interface com shell gráfica (X-Windows)**
- **Comandos típicos**
  - Manipular directorias `mkdir / rmdir / cd`
  - Listar directorias `ls (ls -a ; ls -e; etc)`
  - Ver um ficheiro `more / cat`
  - Copiar um ficheiro `cp`
  - Edição `ed / vi / emacs`
  - Acesso a disco (tape) `tar`
  - Exemplo: `ls -al .[a-z]* | awk '/^[^d]/ {total+=54 } END { print "TOTAL : " total }`  
Mostra o espaço atribuído em disco a todos os ficheiros escondidos cujo nome começa com uma letra minúscula

22

## Unix - Conceitos fundamentais

Sistemas Operativos

- **Sistema de ficheiros**
  - Hierárquico (similar ao ms-dos)
  - Serve de suporte a grande parte do I/O e comunicação entre processos
  - Cada ficheiro tem 9 flags para gerir as protecções:
    - privilégios de Read, Write, Execute para Word, Group, User
  - Existe uma estrutura standard de directorias: `/bin`, `/usr`, `/dev`, etc.
- **Modularidade**
  - O *kernel* do Unix é relativamente pequeno
  - As diversas funções são implementadas por programas "independentes", que correm como processos separados
  - A configuração do sistema e tabelas necessárias para o seu funcionamento são guardados sob a forma de ficheiros de texto editáveis por um processador de texto.
  - Os comandos externos são guardados em directorias fixas, como `/bin`; `/usr/bin`; `/etc`

23

## Unix - Conceitos fundamentais

Sistemas Operativos

- **Processos**
  - São descritos por uma *imagem* que guarda o Código, Dados, e Stack, e é referenciada por um *process identifier-pid*.
  - Cada um tem um certo conjunto de *privilégios* de acesso.
  - Um processo pode lançar processos-filhos que herdam os privilégios do pai (através de *forks*)
- **Comunicação**
  - Através de *pipes* que se comportam como ficheiros.
  - Através de *mailboxes* semelhantes ao modelo tradicional de produtor/consumidor
  - Através de *sockets* que são semelhantes a pipes bidireccionais e que podem funcionar entre máquinas diferentes.
  - Memória partilhada

24

# Sistemas Operativos

V.1.3 V.Lobo 2005

## UNIX - Utilização

Sistemas Operativos

- **Directorias**
  - /bin Comandos do sistema operativo
  - /dev Device drivers  
Um periférico pode ter mais que 1 device driver ! ficheiros tty, fd, hd
  - /etc (e etc...) Ficheiros e comandos de configuração (e  
passwd - contém a informação de login dos utilizadores  
termcap - contém as características dos diferentes terminais
  - /usr Directoria dos utilizadores
  - /usr/bin Programas de aplicação dos utilizadores
  - /usr/man Manuais dos programas
  - /usr/NOME directoria privada do utilizador NOME
  - /home/NOME directoria privada do utilizador NOME
  - /tmp Ficheiros temporários
  - outras: /lost+found, /sys, /lib, /include, /usr/include, /usr/spool, etc

25

## UNIX - Utilização

Sistemas Operativos

- **ls (list files)**
  - Mostra o conteúdo de uma directoria
  - ls -l (long) mostra toda a informação sobre os ficheiros
  - ls -a (all) mostra os ficheiros escondidos (começam com .)
- **cd (change directory)**
  - Muda de directoria (cd - regressa à directoria "home" do utilizador)
- **mkdir / rmdir (make directory/remove directory)**
  - Cria/apaga uma directoria
- **rm (remove file)**
  - Apaga um ficheiro
- **cp (copy)**
  - Copia um ficheiro
- **cat (concatenate)**
  - Mostra o conteúdo de um ficheiro

Exercício:

- Vá à directoria /etc, e veja o ficheiro passwd
- Crie uma subdirectoria na sua área, e copie para lá o ficheiro passwd
- Use o editor "joe" para alterar o seu user-name
- Tente copiar o seu ficheiro para /etc
- Apague tudo o que criou

26

## UNIX - Utilização

Sistemas Operativos

- **chmod**
  - Muda as protecções de um ficheiro. Pode-se indicar as novas protecções especificando os bits de protecção, ou simbolicamente ( p.ex: w+r = Word pode ler)
- **mount**
  - permite integrar um filesystem de um periférico (disco, disquete, cdrom), no filesystem do sistema. O dispositivo montado aparece então como uma subdirectoria
- **mail**
  - programa para enviar/receber correio
- **finger**
  - fornece informação sobre um dado utilizador

27

## UNIX - Utilização

Sistemas Operativos

- **Ficheiros escondidos na área do utilizador**
  - guardam a configuração de diversos programas
  - .login parâmetros da C-shell (aliases, history, etc)
  - .plan dados para o programa finger
  - .mailrc ficheiros relacionados com mail
  - .newsrc ficheiros relacionados com news
- **Exercícios**
  - Crie uma directoria com um ficheiro protegido (rwe só para si), outro aberto para todos (rwe para world), e um terceiro com acesso só de leitura para outros (r-e para world)
  - Tente ler os ficheiros dos seus camaradas
  - Mande mensagens de correio para os seus camaradas
  - Faça "talk" para um dos seus camaradas

28

## Unix- Funções de sistema

Sistemas Operativos

- **Processos**
  - Criação *fork*
  - Execução de um programa *exec*
  - Auto-terminação *exit*
  - Terminação de outro processo *kill*
  - Esperar pela terminação de um filho *wait*
- **Pipes**
  - Criação *pipe mknod*
  - Associação *open*
  - Leitura *read*
  - Escrita *write*
  - Fichar *close*

29

## Unix- Funções de sistema

Sistemas Operativos

- **Semáforos (SystemV)**
  - Criação *semget*
  - Operações *semop*
  - Controlo *semctl*
- **Mailboxes (SystemV)**
  - Criação *msgget*
  - Leitura *msgrcv*
  - Escrita *msgsnd*
  - Controlo *msgctl*
- **Memória partilhada (SystemV)**
  - Criação *shmget*
  - Mapear *shmat*
  - Retirar *shmdt*

30

# Sistemas Operativos

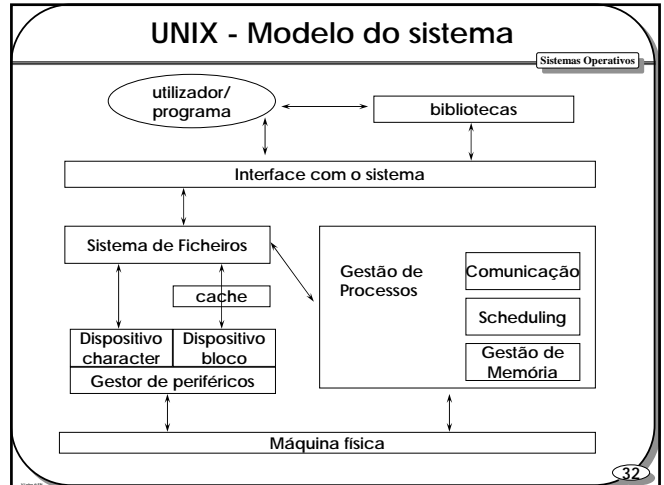
V.1.3 V.Lobo 2005

## Unix- Funções de sistema

Sistemas Operativos

- Sockets (BSD)
  - Criação *socket*
  - Associação a um nome *bind*
  - Ligação *listen*
  - connect*
  - Envio *accept*  
*send*  
*write*  
*sendto*
  - Recepção *recv*  
*read*  
*recvfrom*

31



## Unix - Administração do sistema

Sistemas Operativos

- Qualquer sistema UNIX necessita de um "gestor de sistema", que tem privilégios de SUPERUSER
  - O username do Superuser é "root"
  - A "root" tem todos os privilégios
  - Pode-se ter privilégios de Superuser temporário com o comando SU
- Atribuições do gestor de sistema
  - Gerir utilizadores
    - Criar utilizadores, apagar utilizadores que deixaram o sistema
  - Gerir hardware
    - Configurar os device drivers do hardware de modo a tirar o máximo partido deste
  - Gerir software
    - Instalar programas (normalmente é necessário compilá-los), garantir que não há conflitos

33

## Unix - Administração do sistema

Sistemas Operativos

- Atribuições do gestor de sistema (cont.)
  - Fazer backups
    - MUITO IMPORTANTE !!! É necessário ter 2 conjuntos de backup, para o caso de haver uma falha ao fazer o backup. Pode-se ter um backup completo muito espaçado e fazer apenas backups incrementais
  - "Limpezas diárias"
    - Retirar os LOGFILES que são gerados pelas diversas aplicações, ficheiros temporários perdidos, etc.
  - "Troubleshooting"
    - Detectar e resolver bugs que apareçam, ajudar os utilizadores com os seus problemas, etc.
  - Manter documentação
    - Ter manuais dos programas utilizados (se possível on-line e em papel), documentar MUITO BEM a configuração do sistema: endereços usados, directorias, etc.

34

## Unix - Administração do sistema

Sistemas Operativos

- Segurança
  - Verificar que não há brechas de segurança: verificar os logfiles para detectar anomalias, instalar firewalls para limitar o acesso exterior ao sistema, etc
- Contabilidade
  - Monitorar a utilização dos recursos (disco, cpu, impressora, etc), de modo a não deixar degradar a performance do sistema. Propor actualizações de hardware/software. Fazer uma atribuição de custos se necessário.

35