

# Sistemas Operacionais

Prof. Me. Pietro M. de Oliveira

## Unidade V

Sistemas Operacionais Multimídia

Escalonamento.

Sistemas de Arquivos.

Alocação de Arquivos.

Máquinas Virtuais

Hipervisores.

Linux.

Scripts.

Arquivo TXT:

Não possui a dimensão “tempo”.

Arquivo MP3:

Acessado em ritmo próprio.

Sistemas multimídia

Novos tipos de sistemas de arquivos.

Novas exigências para o  
escalonador e outras partes do SO.

Arquivos com subarquivos.

Exemplo: DVD.

SOs com suporte a vários dispositivos de mídia

Áudio, vídeo, texto.

Arquivos Multimídia

Vários arquivos em um só.

Ex.: vídeo, áudio e legenda.

Ritmo de acesso específico:

Carregamento fracionado.

Escalonador tem função vital.

Geralmente um arquivo TXT é carregado por completo em memória.

Um arquivo multimídia tende a ser carregado à medida que é executado.

Escalonador de um SO multimídia:

- Garantir a reprodução das mídias.

- Não abandonar outros processos.

Vários processos devem ser executados

Um processo não pode monopolizar a CPU.

*Round-Robin:*

Processo executado de tempos em tempos.

Dados são salvos em buffers.

Temporizador – sincronização.

## Armazenamento com garantia de performance

**Vazão:** taxa de dados maximizada.

**Demora:** tempo entre solicitação do usuário e reprodução minimizado.

**Tremulação:** demora na reprodução do fluxo de dados minimizada.

**Confiabilidade:** tolerância a erros maximizada.

Arquivos convencionais são carregados por completo.

Arquivos multimídia:

- Carregar blocos mediante a solicitações

  - Ex.: ler blocos de conforme a necessidade.

  - Maior controle no ponto de acesso.

  - Gasta mais recursos.

- Carregar blocos sem depender de requisições.

  - Dados são utilizados continuamente.

  - Controle no ponto de acesso é reduzido.

  - Gasta menos recursos computacionais.



## Servidores multimídia

- Único ou múltiplos discos.

- Alguns arquivos são mais acessados que outros.

## Disco único

- Agrupar arquivos mais acessados.

- Algoritmos de tubo de órgãos & *caching*.

- Diminuir o trabalho do HD.

## Múltiplos discos

- Armazenar blocos em vários discos.

- Algoritmos que garantam balanceamento.

# Sistemas de Arquivos Multimídia



Fonte: [sxc](#) / [freepik.com](#)

“Transformar” um computador em vários  
Múltiplas máquinas com múltiplos S.Os.  
S.Os com diferentes tipos de processamento.

## Máquina Virtual (virtualização)

Duplicata da máquina real.  
Economia de hardware, espaço físico e energia.  
Criar ambientes específicos.

## Tipos de Máquinas Virtuais

De processo: estende os tipos de processos que podem ser executados na máquina física.

*Java Virtual Machine (JVM)*

De SO: executa um SO no mesmo hardware, como se fosse um novo PC fosse ligado.

VMWare

De sistema: emula um hardware completo, incluindo também um novo SO.

VMWare

**De processo:** estende os tipos de processos que podem ser executados na máquina física.

*Java Virtual Machine (JVM), Dalvik (Android).*

**De SO:** executa um SO no mesmo hardware, como se fosse um novo PC fosse ligado.

VMWare, ESXi, FreeBSD Jails.

**De sistema:** emula um hardware completo, incluindo um novo SO.

VMWare, VirtualBox.



## Monitor de Máquina Virtual

Aplica técnicas de controle de virtualização.

### Tipo 1

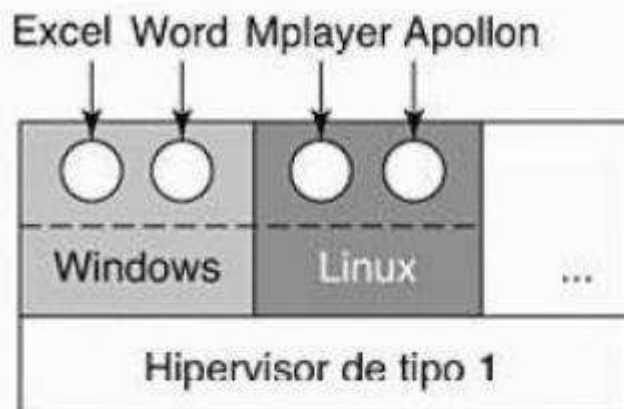
Ligado diretamente ao hardware (Servidor).

SOs prontos para uso.

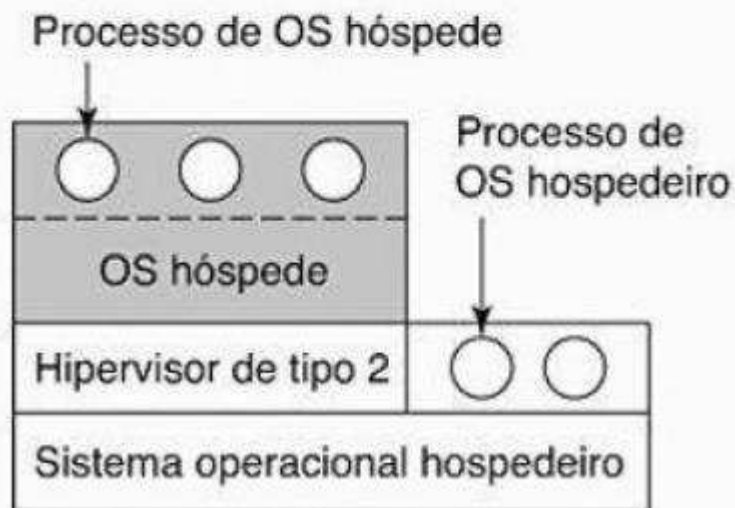
### Tipo 2

Software sobre o SO  
original (Cliente)

Processos do SO original  
executando simultaneamente.



(a)



(b)

(a) Hipervisor de tipo 1; (b) Hipervisor de tipo 2  
 Fonte: Tanenbaum (2010, p 42)

Criado na Finlândia, por Linus Torvalds (1991)

Baseado no MINIX (Tanenbaum).

UNIX.

Fins acadêmicos.

Código disponível ao público.

Licença:

Software Livre – OpenSource.

GNU General Public License (GPL).



Público contribui com implementações.

- Correção de falhas.

- Novas funcionalidades.

- Novas distribuições.

- Aumento de segurança e confiabilidade.

Kernel (corpo principal do SO)

- Não comercializável.

Atualmente, diversas distribuições:

- Ubuntu, Slackware, Red Hat

- Interface gráfica (GUI)

Como Instalar?

Primeiro Contato

Interface Gráfica.

Shell (comandos de texto).

Estrutura de Pastas.

Desenvolvimento.

Scripts.

GCC - programação em C.

**Shell:** interpretador de comandos do usuário.

Não faz parte do kernel (programa à parte).

Funciona como o próprio SO

Executar programas.

Manipular arquivos.

Monitorar processos.

Automatizar tarefas.

Roda no modo usuário (a princípio)

Vários tipos:

SH, CSH, KSH, TCSH, BASH.

## Comandos básicos:

- Manipulação de arquivos e diretórios.

- Controle de processos.

- Compilar código em C.

- Executar scripts.

- Comandos de cursor.

Diversos tutoriais online.

Ajuda sobre como executar o terminal

help

Manual de comandos

man [comando]

Comandos:

ls – lista o conteúdo do diretório corrente

cd – muda de diretório

pwd – imprime o diretório corrente

clear – limpa a tela

## Comandos (continuação):

ps – lista os programas em execução

top – monitor de processos

kill – finaliza um processo

mkdir – cria um diretório

rmdir – remove um diretório

rm – remove um arquivo

cp – copia arquivos/diretórios

mv – move ou renomeia arquivos/dir.

## Comandos (continuação):

tail – mostra o final do conteúdo do arq.

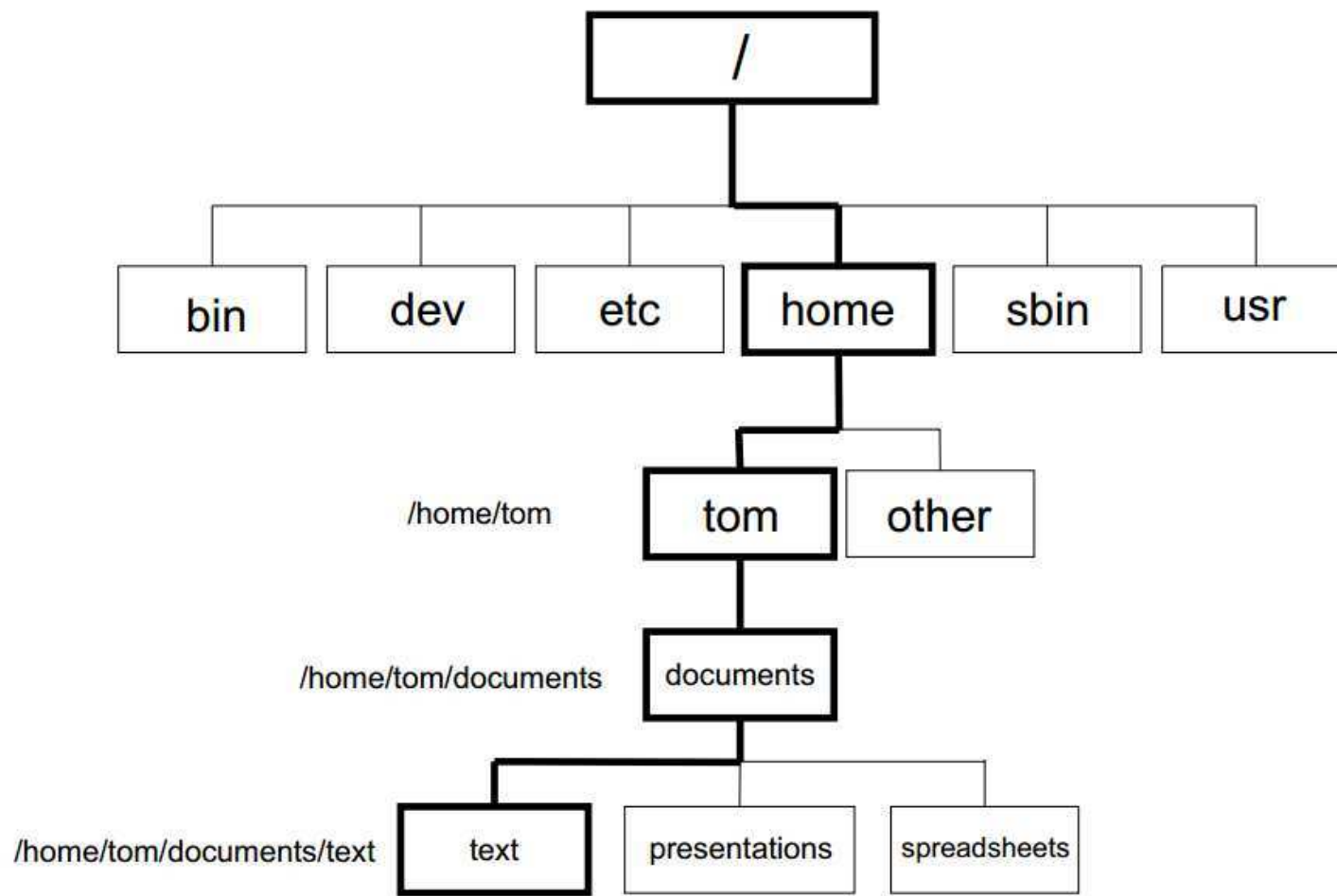
chmod – muda as permissões de um arq.

echo – imprime uma mensagem na tela

## Caminho:

Relativo - leva em consideração o diretório corrente. Ex.: ***./caminho***

Absoluto - desde o diretório raiz. Ex.: ***/home/caminho***





Shell scripts:

“Programas” com comandos unix

Pode-se utilizar:

- Comandos unix.

- Variáveis.

- Estruturas de controle/repetição.

- Funções

- “Subscripts”

- Executáveis

Semelhantes aos batch (.bat)

Exemplo de declaração de variáveis:

```
variavel="teste"
```

```
echo $variavel
```

```
resultado: teste
```

Comandos de controle de fluxo

```
if [ condição ]
```

```
then comando1
```

```
else comando2
```

```
fi
```

## Controle de fluxo (continuação)

```
case condição in
```

```
    opção1) comando1.0; comando1.1;;
```

```
    opção2) comando2.0; comando2.1;;
```

```
    *) comandoX;;
```

```
esac
```

## Controle de fluxo (continuação)

```
variavel="valor"  
while [ $variavel = "valor" ]; do  
    comando1  
    comando2  
done
```

## Controle de fluxo: outras opções

until:

“inverso” do **while**

for:

desde – até (de – para)

útil para percorrer uma lista ou arquivo

funções:

blocos de comandos repetitivos

## Janelas Gráficas:

Xdialog

Sistemas multimídias

- Escalonador multimídia.

- Sistemas de arquivos multimídia.

Máquinas virtuais

- Hipervisores

Linux

- Shell, comandos, scripts (tutorial).

# Sistemas Operacionais

Prof. Me. Pietro M. de Oliveira