Professor
Edson Pedro FERLIN

Sistemas Operacionais

Tipos de Sistemas Operacionais

Prof. Edson Pedro Ferlin

Agradecimento ao Prof. Osmar Betazzi Dordal

Professor
Edson Pedro FERLIN

Sistemas Operacionais

- Objetivos
  - Apresentar os tipos de sistemas operacionais
- Conteúdos
  - Interação
  - Tipos de SO
  - Estrutura dos SO

Definição de Sistema Operacional

- Um Sistema Operacional é um programa ou um conjunto de programas inter-relacionados, cuja finalidade é agir como:
  - Intermediário (***interface***) entre o usuário e o hardware (**Aplicação de Usuário e Hardware**);
  - Gerenciador de recursos:
    - Impressão;
    - Arquivos;
    - Memória.

Vantagem do Sistema Operacional

- Eficiência na utilização dos recursos de hardware.
- Flexibilidade da utilização da máquina – evolução.
- Uso compartilhado e protegido dos diversos componentes de hardware por diversos usuários.

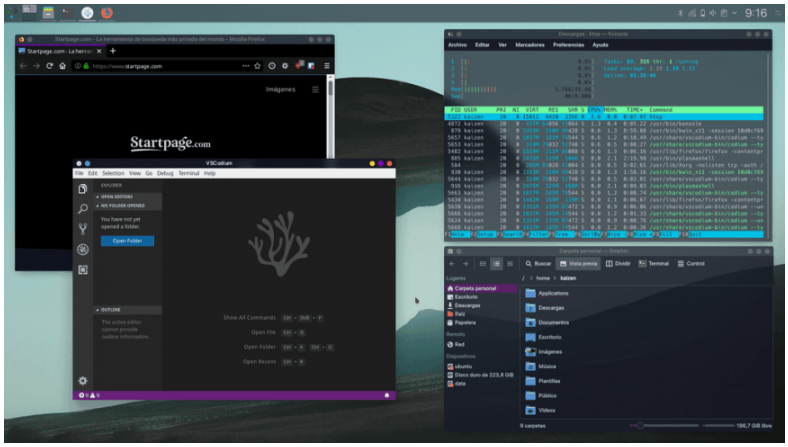
Interação com o Sistema Operacional

- Linguagem de Comando – Texto

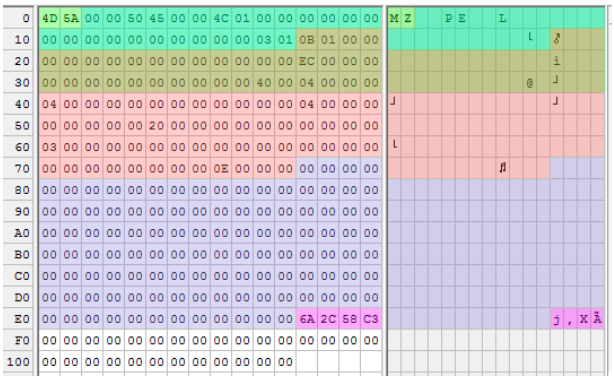
```
compgen [-abcdefgjksuv] [-o option]  >  return [n]
complete [-abcdefgjksuv] [-pr] [-DE] >  select NAME [in WORDS ... ;] do COMM>
compopt [-o|+o option] [-DE] [name ..>  set [-abefhkmnptuvxBCHP] [-o option->
continue [n]                            shift [n]
coproc [NAME] command [redirections]    shopt [-pqsu] [-o] [optname ...]
declare [-aAfFgilrtux] [-p] [name[=va>  source filename [arguments]
dirs [-clpv] [+N] [-N]                  suspend [-f]
disown [-h] [-ar] [jobspec ...]         test [expr]
echo [-neE] [arg ...]                   time [-p] pipeline
enable [-a] [-dnps] [-f filename] [na>  times
eval [arg ...]                          trap [-lp] [[arg] signal_spec ...]
exec [-cl] [-a name] [command [argume>  true
exit [n]                                type [-afptP] name [name ...]
export [-fn] [name[=value] ...] or ex>  typeset [-aAfFgilrtux] [-p] name[=va>
false                                   ulimit [-SHacdefilmnpqrstuvx] [limit>
fc [-e ename] [-lnr] [first] [last] o>  umask [-p] [-S] [mode]
fg [job_spec]                           unalias [-a] name [name ...]
for NAME [in WORDS ... ] ; do COMMAND>  unset [-f] [-v] [name ...]
for (( exp1; exp2; exp3 )); do COMMAN>  until COMMANDS; do COMMANDS; done
function name { COMMANDS ; } or name >  variables - Names and meanings of so>
getopts optstring name [arg]            wait [id]
hash [-lr] [-p pathname] [-dt] [name >  while COMMANDS; do COMMANDS; done
help [-dms] [pattern ...]               { COMMANDS ; }
```


Interação com o Sistema Operacional

- Forma Gráfica



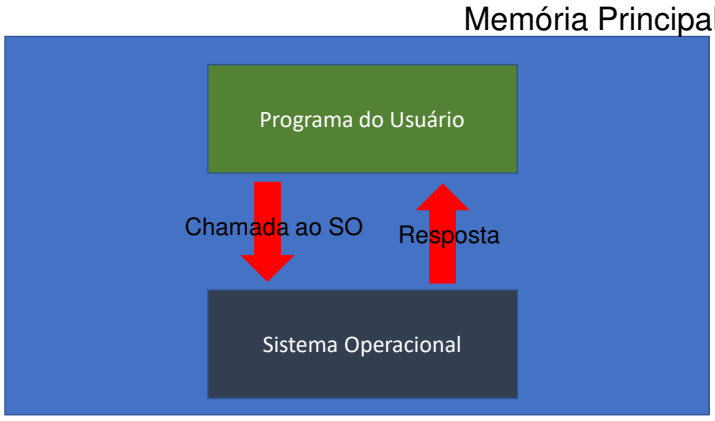
Estrutura – Arquivo EXE




```
0x6A  // PUSH
0x2C  // value to push
0x58  // POP EAX
0xC3  // RETN
```

268 bytes is the absolute minimum size for a working executable under Windows 7 64-bit edition

Interação com o Sistema Operacional



Interação com o Sistema Operacional

- **Chamadas de Sistema – *Systems calls.***
  - Permitem um controle mais eficiente sobre as operações do sistema.
- Cada Sistema Operacional tem suas chamadas de sistema.
- Algumas chamadas são têm nomes iguais ou bem próximos.

Chamada de Sistema Unix

pid = fork() - Essa chamada cria um processo (filho), idêntico ao pai
pid = waitpid(pid, &statloc, options) - O processo pai volta a rodar somente após o término do filho, ou seja, o processo pai aguarda o filho ser finalizado
s = execv(name, argv, enrionp) - É realizado, no processo, a substituição da imagem do núcleo
exit(num) - Termina um processo e retorna um status (um número identificado algo, como sucesso, erro etc)
fd = open (file, abertura ...) - Abre um arquivo (leitura, escrita ou os dois)
s = close(fd) - Fecha o arquivo anteriormente aberto
n = read (fd, buffer, nbytes) - Lê dados do arquivo 'fd' no 'buffer', de 'nbytes' valores de bytes
n = write(fd, buffer, nbytes) - Analogamente ao read, mas escreve
position = lseek(fd, offset, whence) - Move o ponteiro de 'fd'
s = stat(name, &buf) - Retornar informações do arquivo
s = chdir(nome) - Altera o diretório em que se está trabalhando
s = kill(pid, sinal) - Envia um sinal 'sinal' para o processo de PID 'pid'
seconds = time(&seconds) - Essa chamada retorna o tempo, em segundos, desde 1o de janeiro de 1970
s = chmod(nome, modo) - Altera o sistema de proteção do arquivo 'nome'
s = mkdir(nome, modo) - Cria uma pasta (diretório)
s = rmdir(nome) - Apaga o diretório de nome "nome", previamente criado
s = mount(special, nome, flag) - Cria um sistema de arquivos (como um dvd ou pendrive)
s = umount(special) - Desmonta o sistema de arquivos previamente criado
s = link(nome1, nome2) - Cria a entrada 'nome2' apontando pra 'nome1'
s = unlink(nome) - Desfaz a entrada 'nome'

Chamada de Sistema Windows

CreateProcess - fork
WaitForSingleObject - waipid
ExitProcess - exit
CreateFile - open
ReadFile - read
WriteFile - write
CloseHandle - close
SetFilePointer - lseek
GetFileAttributesEx - stat
CreateDirectory - mkdir
RemoveDirectory - rmdir
SetCurrentDirectory - chdir
GetLocalTime - time
DeleteFile - unlink

Formas de Processamento do Sistema Operacional

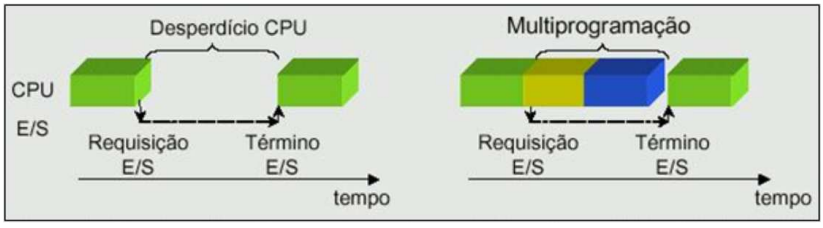
- **Monoprogramação** – Serial.
  - Recursos alocados a um **único** programa.
- **Multiprogramação** – Concorrente.
  - Recursos **dinamicamente** reassociados entre uma **coleção** de programas em diferentes estágios.

Sistemas Operacionais Monoprogramáveis – Monotarefa

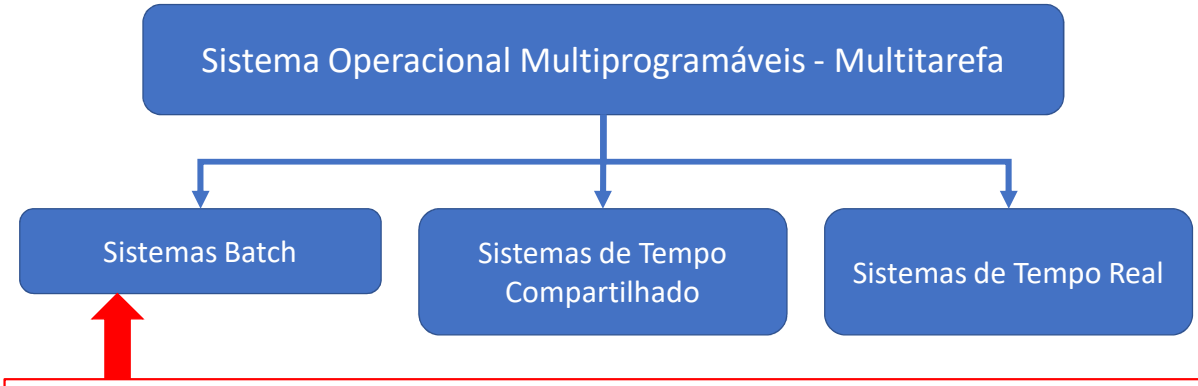
- Sua característica está em permitir que o **processador**, a **memória** e os **periféricos** permaneçam **exclusivamente dedicados** à execução de um único programa.
- Os **recursos** são **mal utilizados**, mas em contrapartida, são implementados com facilidade.
- O processo só tem três estados e só poderá estar em um desses:

Sistemas Operacionais Multiprogramáveis – Multitarefa

- **Mantêm diversos programas em memória simultaneamente**.
- **Diversas tarefas** em um **único processador**
  - Enquanto **um** programa **executa outros esperam**.
- Demanda um mecanismo de trocas rápidas de processos
  - Maior otimização.

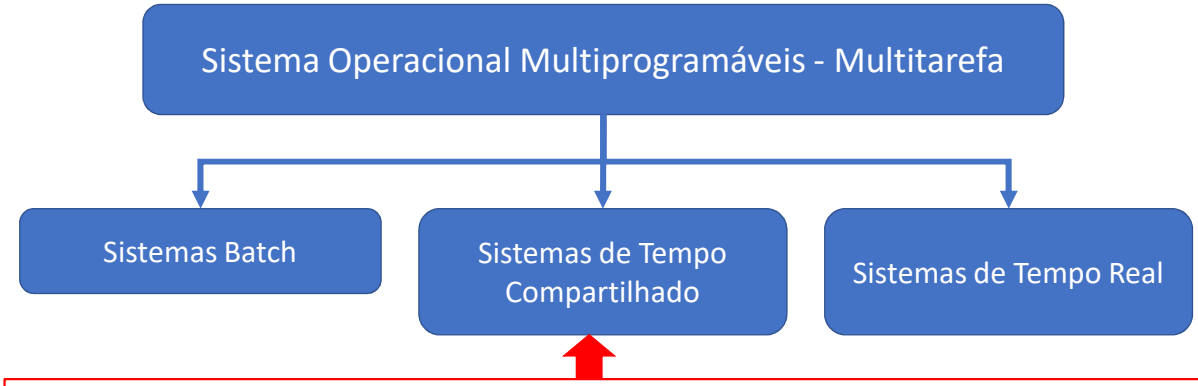


Sistemas Operacionais Multiprogramáveis – Multitarefa



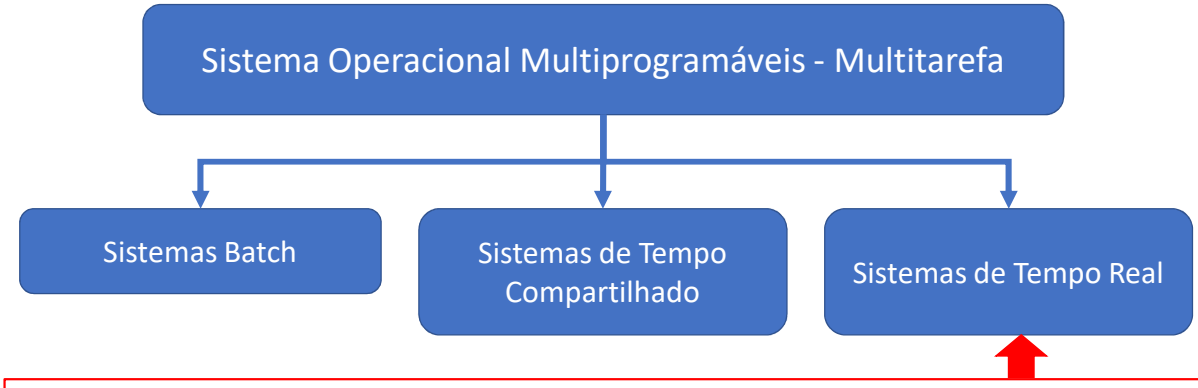
- Executa de uma vez (lote);
- São executados em ordem sequencial;
- Não existe interação entre usuário e tarefa (executa até o fim).

Sistemas Operacionais Multiprogramáveis – Multitarefa



- O sistema possibilita que os usuários interajam com suas computações na forma de diálogo
- Pode ser projetado como sistema monousuário ou multiusuário usando conceito de multiprogramação e *time-sharing (mainframe).*

Sistemas Operacionais Multiprogramáveis – Multitarefa



- Sistemas com tempo de resposta predefinidos;
- Exemplos: equipamentos médicos; freios ABS; sistemas de navegação; etc.

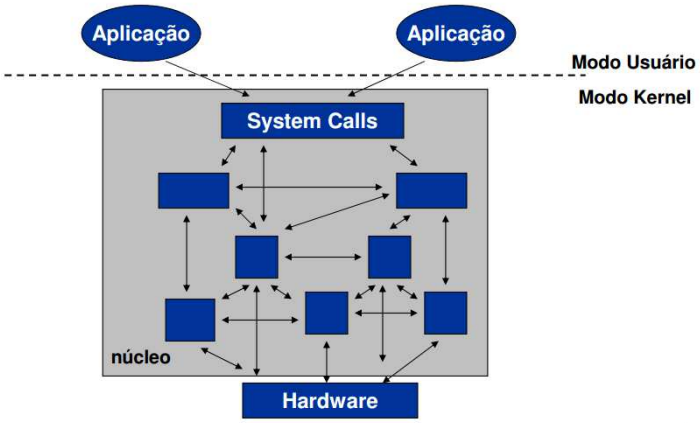
Estrutura dos Sistemas Operacionais

- Sistemas Operacionais são normalmente grandes e complexas coleções de rotinas de software.
- Projetistas devem dar grande ênfase a sua organização interna e estrutura:
  - Monolítico;
  - Micronúcleo;
  - Camadas;
  - Máquina Virtual.

Estrutura dos Sistemas Operacionais

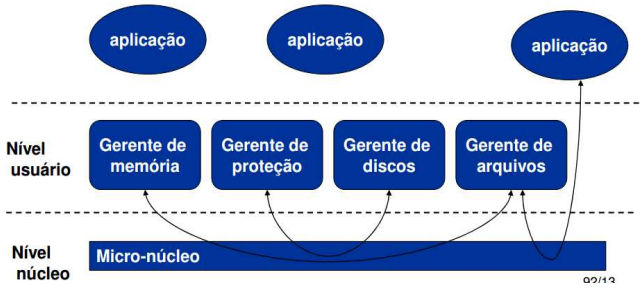
- Projetistas devem dar grande ênfase a sua organização interna e estrutura:
  - Monolítico;
  - Micronúcleo;
  - Camadas;
  - Máquina Virtual.

MS-DOS
Windows
Unix e Linux



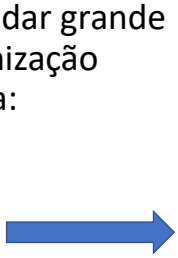
Estrutura dos Sistemas Operacionais

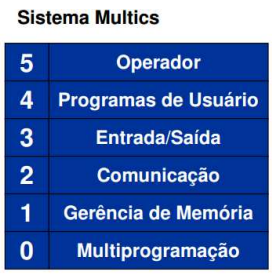
- Projetistas devem dar grande ênfase a sua organização interna e estrutura:
  - Monolítico;
  - Micronúcleo;
  - Camadas;
  - Máquina Virtual.

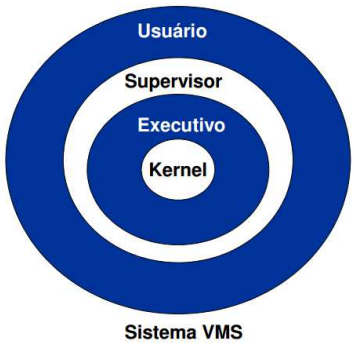


Estrutura dos Sistemas Operacionais

- Projetistas devem dar grande ênfase a sua organização interna e estrutura:
  - Monolítico;
  - Micronúcleo;
  - Camadas;
  - Máquina Virtual.



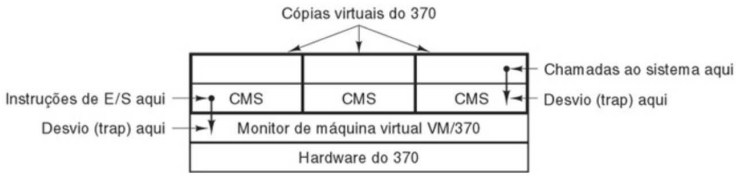




Sistema *MULTICS*
OpenVMS

Estrutura dos Sistemas Operacionais

- Projetistas devem dar grande ênfase a sua organização interna e estrutura:
  - Monolítico;
  - Micronúcleo;
  - Camadas;
  - Máquina Virtual.



Estrutura do VM/370 com o CMS
Exemplo: DOS no Windows, JVM, etc.

Contato

eferlin@live.com

(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) ProfEdsonPedroFerlin