UNIVERSIDADE
EDUARDO
MONDLANE

FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA
LICENCIATURA EM ENGENHARIA INFORMÁTICA
SISTEMAS OPERATIVOS E PROGRAMAÇÃO CONCORRENTE

Processos

Docente:

Engº. Délcio Chadreca (MsC)

Dr. Alfredo Covele (MsC)

Tópicos da Aula

- Programa Interface: Shell ou GUI
- Niveis de Operação
- Chamadas de Sistema
- Processos

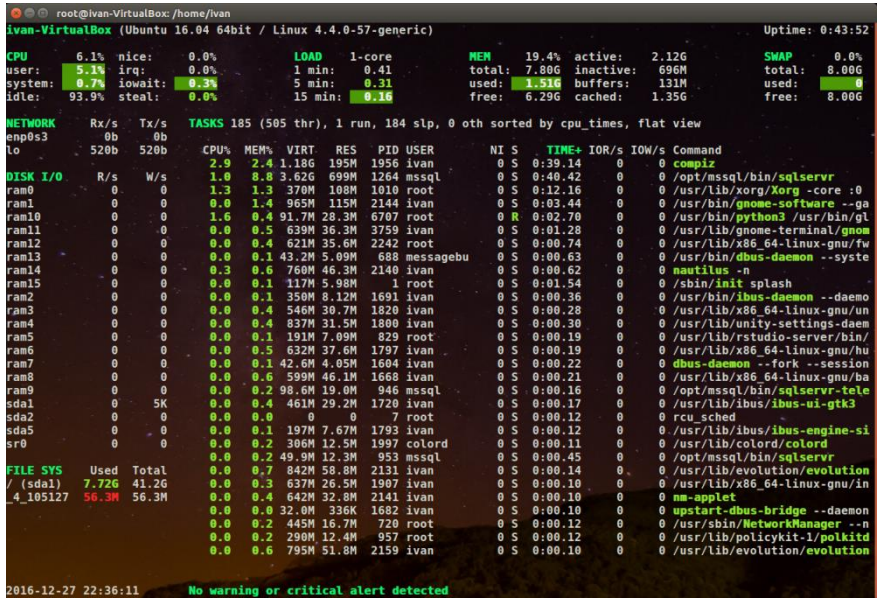
Programa Iterface: Shell ou GUI

- Programa que permite a interação usuário-máquina:
  - ***Shell*** (ou **Interpretador de comandos**):
    - Programa baseado em texto.
  - ***GUI*** (*Graphical User Interface*):
    - Interface gráfica com o usuário através de ícones

Shell vs GUI

- Facilidade
- Control
- Multitarefas
- Velocidade

- Recursos
- Scripting
- Acesso Remoto
- Tensão visual



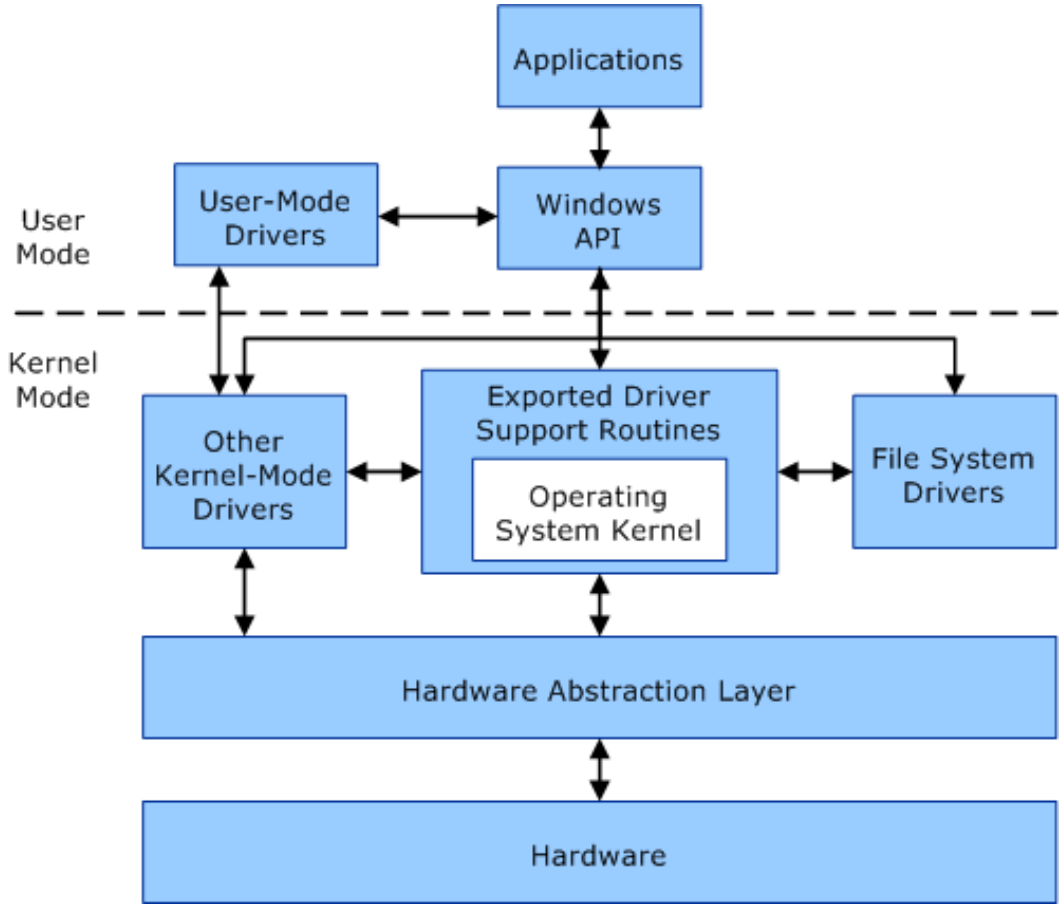
Nives de Operação

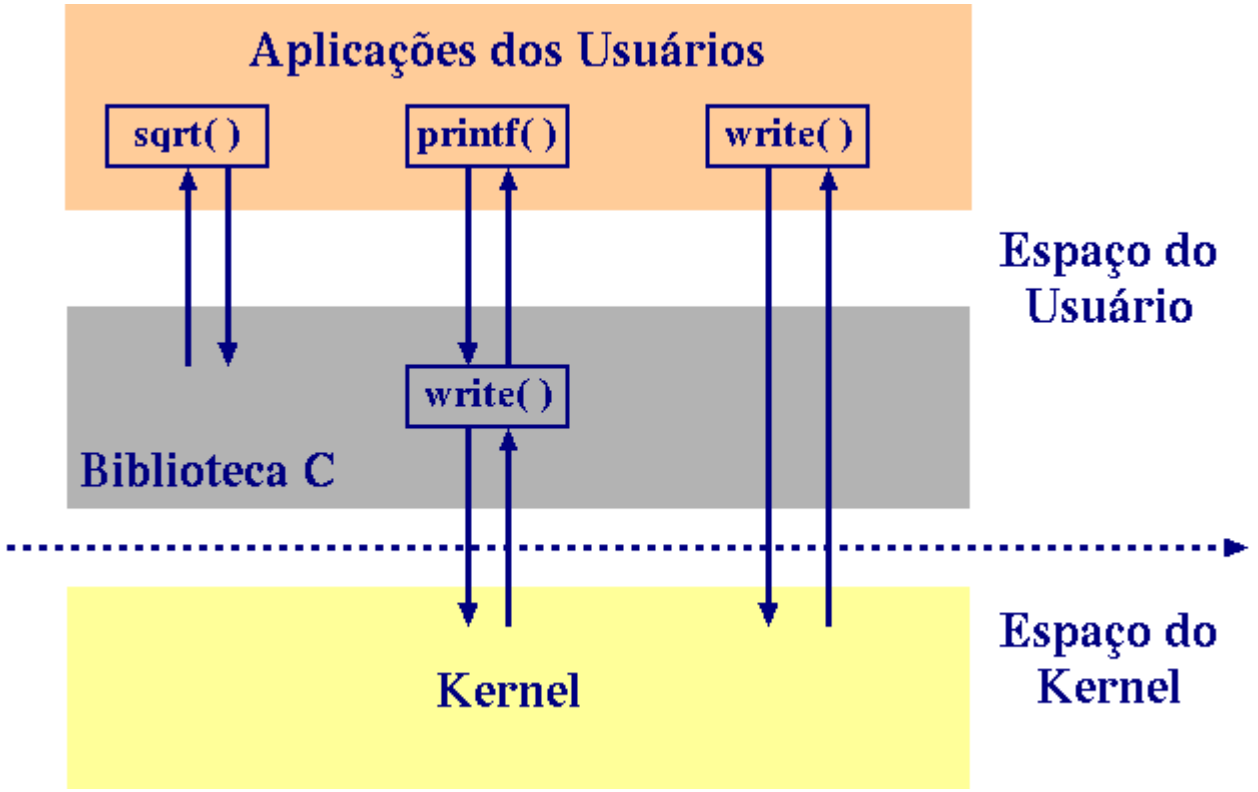
- A maioria dos computadores tem dois níveis de operação:
  - **Modo núcleo** (também chamado **modo *kernel*** ou **modo supervisor**).
  - **Modo usuário.**

Cont.

- **Modo núcleo** (ou **modo *kernel*** ou **modo supervisor**):
  - Nível de operação do SO.
  - Nesse modo o SO tem acesso completo a todo hardware e pode executar qualquer instrução que a máquina seja capaz de executar.

- **Modo usuário**:
  - Apenas um subconjunto de instruções da máquina está disponível.
  - As instruções que afetam o controle da máquina ou realizam E/S (Entrada/Saída) são proibidas.


Applications
User-Mode Drivers
Windows API
User Mode
Kernel Mode
Other Kernel-Mode Drivers
Exported Driver Support Routines
Operating System Kernel
File System Drivers
Hardware Abstraction Layer
Hardware


Aplicações dos Usuários
sqrt( )
printf( )
write( )
Espaço do Usuário
write( )
Biblioteca C
Kernel
Espaço do Kernel

Chamada de Sistema

- A interface entre o S.O. e os programas de usuário é definida pelo conjunto de instruções estendidas que o sistema operacional proporciona.
- Essas instruções são conhecidas como chamadas de sistema (*system calls*) → manipulam diversos objetos geridos pelo S.O.

Chamadas de Sistema:

- Pode-se dizer que são métodos/funções utilizados para que os programas de usuários solicitem serviços providos pelo sistema operacional.
- Cada chamada corresponde a um procedimento de uma biblioteca contida no sistema operacional.
- As chamadas de sistema executam em modo protegido, sem a interferência do usuário.

Processo de Execução de uma Chamada de Sistema:

- Os serviços são requisitados através da colocação de parâmetros em lugares muito bem determinados (por exemplo, em registradores).
- Segue-se a execução de uma instrução especial de *trap* (Chamada ao Supervisor ou Chamada ao Kernel).
- A execução deste tipo de instrução chaveia a máquina do Modo Usuário para o Modo Kernel, e transfere o controle para o S.O.

Processo de Execução de uma Chamada de Sistema:

- O S.O. examina os parâmetros para determinar qual das chamadas de sistema deve ser executada.
- A seguir, o S.O. verifica em uma tabela indexada, o endereço do procedimento que executa a chamada ao sistema.
- Após a conclusão da chamada de sistema, o controle retorna ao programa do usuário.

APIs que facilitam a chamada de sistema:

- Win32
- Posix
- Java

Passagem de parametros:

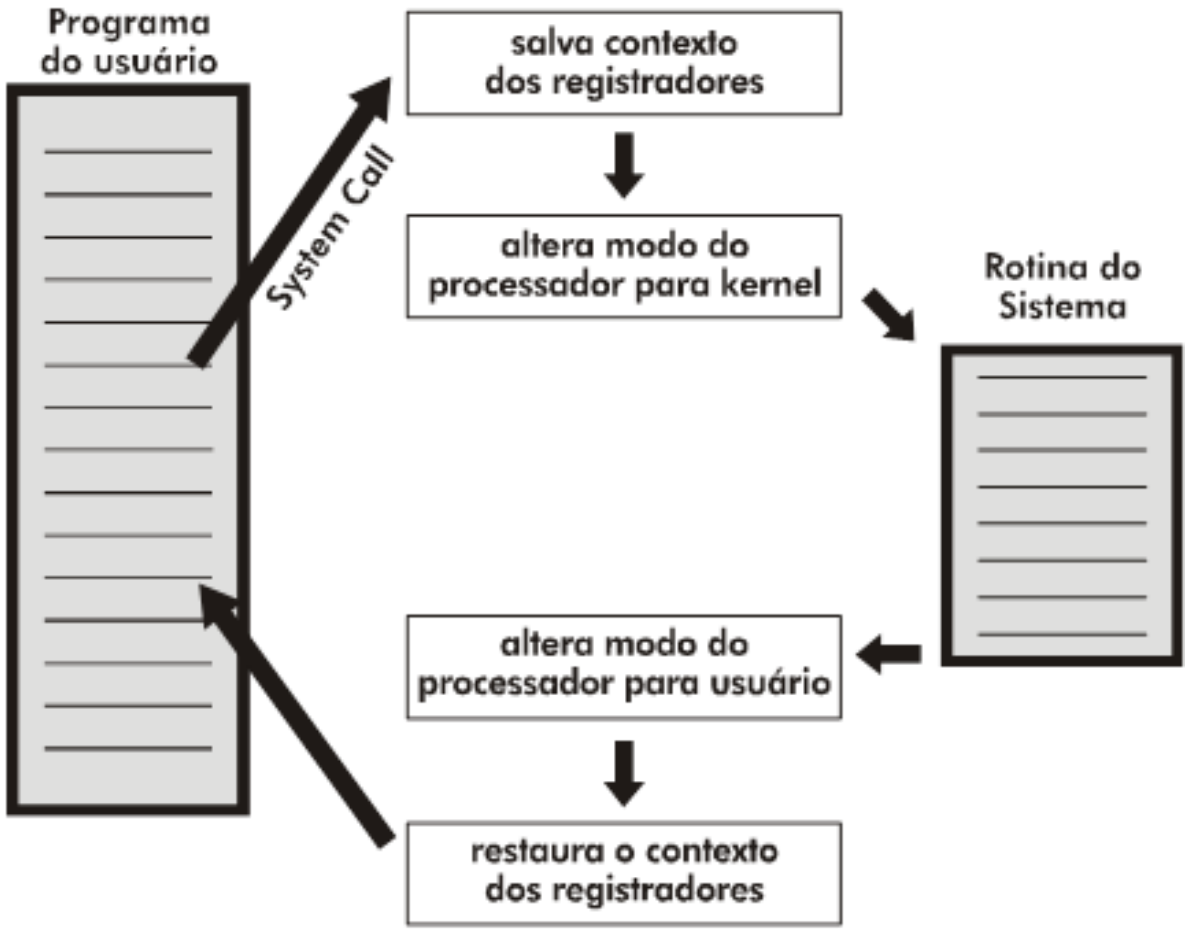
- Registratadores
- Tabela em memoria
- Pilha

Tipos de Chamada Sistema

- Controlo de processos
- Manipulação de arquivos
- Manipulação de dispositivos
- Manipulação de Informações
- Comunicação
- Protecção

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

Operação Chamada Sistema



Sistema multiusuario e multitarefa

Um SO multisuario permite **acessos simulateneoas ao computador**, onde cada usuario tem sua propria conta e ambiente de trabalho isolado.

Multitarefa – e a capacidade do SO alternar rapidamente entre varios processos ou tarefas em execusao, dando a ilusao de que estao correndo em simultaneo

Multiprogramacao

Sistema monoprogramavel permite a execusao de apenas um programa por vez em um computador , nao suporta a execusao simulatanea de multiplos programas (execusao instrucao-a-instrucao, menos eficiente)

Sistema multiprogramavel permite a execusao de varios programas de forma intercalada, mantendo varios programas na memoria e alternado entre eles para ptimizar o uso da CPU

Processo

- Conceito fundamental para todos os SOs.
- Ambiente onde se executa um programa.
- Um processo é basicamente um programa em execução.
- O SO materializa o processo através de uma tabela, chamada **tabela de processos**.

Cont.

Cada entrada da tabela é chamada de ***bloco de controle do processo*** (***P****rocess* ***C****ontrol* ***B****lock* **- PCB**).

Um PCB é responsável por manter todas as informações referentes a um determinado processo nomeadamente:

PCB	Description
Process State	Current, Ready , Running, writing…
process Privileges	Allow or Disallow to access system Resources
Process ID	Unique id for Process
Pointer	Points to parent process
Program Counter	Next instruction to be executed
CPU register	Storing data
CPU Scheduling	Process priority
Memory Management Information	Information of page tables, memory limit, etc
Accounting Information	Amount of CPU used
i/o Status Information	List I/O devices allocated to process

Processo

- Em um sistema multiprogramável, um processo passa por uma série de estados, durante a sua existência.
- Existem três estados em que um processo pode se encontrar no sistema:
  - **Pronto**: um processo já se encontra pronto para executar, entretanto o processador está sendo ocupado por outro processo.
  - **Execução**: um processo está efetivamente ocupando o processador para seu processamento.
  - **Bloqueado/Espera**: um processo está no estado de bloqueio/espera quando aguarda a ocorrência de determinado evento para continuar sua execução.

Bloqueio de Processo

Um processo pode estar bloqueado por dois motivos:

O processo está pronto para executar, porém não há ainda alguma entrada disponível para permitir essa execução.

Exemplo: Um comando está pronto para ser executado, mas precisa de parâmetros de entrada para sua execução.

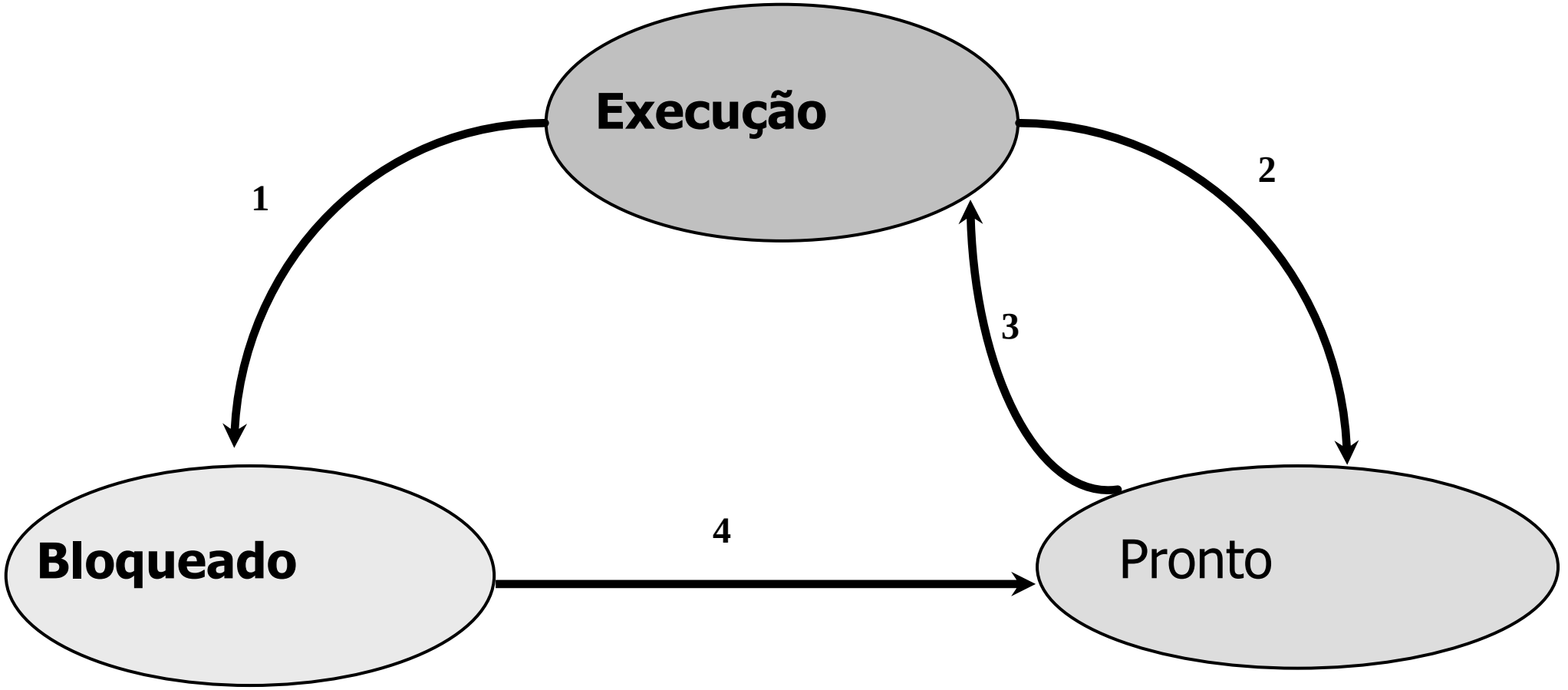
O processo estava processando, necessitou realizar alguma operação de E/S, desocupou o processador, e está aguardando a retomada do processador.

Mudança de Estados

- Um processo muda de estado diversas vezes, durante o seu processamento, em função de eventos gerados por ele próprio (***eventos voluntários***) ou pelo sistema operacional (***eventos involuntários***).

- Basicamente, existem quatro mudanças de estado que podem ocorrer a um processo:

Estados de um processo



Estados de um processo

1. Execução ➔ Bloqueado/Espera:

- Um processo em execução passa para o estado bloqueado ou de espera por eventos gerados pelo próprio processo.
  - **Exemplo**: Espera por uma operação de E/S.

2. Execução ➔ Pronto:

- Um processo em execução passa para o estado de pronto por eventos gerados pelo sistema.
  - **Exemplo**: Fim da fatia de tempo (*time-slice*) que o processo possui para sua execução.

3. Pronto ➔ Execução:

- Processo que estava na lista de processos pronto passa a ser executado pelo processador.
- Quando um processo é criado, o sistema o coloca em uma lista de processos no estado pronto, onde aguarda uma oportunidade para ser executado, a partir do momento que “ganha” a CPU passa para o estado de execução.

4.Bloqueado/Espera ➔ Pronto:

- Um processo no estado de espera passa para o estado de pronto quando a operação solicitada ou o recurso é atendido.
- Um processo no estado de espera sempre terá que passar pelo estado de pronto antes de ser executado novamente. Não existe a mudança do estado bloqueado para o estado de execução diretamente.

Diagrama de estados de um processo

Estado Terminado

Um processo pode passar para o estado terminado por razões como:

- Término normal de execução;
- Eliminação por um outro processo;
- Eliminação forçada por ausência de recursos disponíveis no sistema.

Trabalho de Pesquisa Complementar

- Sinais
- Processos Co-operação
- Memoria Partilhada
- Mensagem Baseada em IPC

Nota: Os presentes tópicos são complementares a matéria sobre processos no seu todo, o estudante deverá fazer uma pesquisa e leitura individual/grupo sobre os mesmos.

OBRIGADO !!!