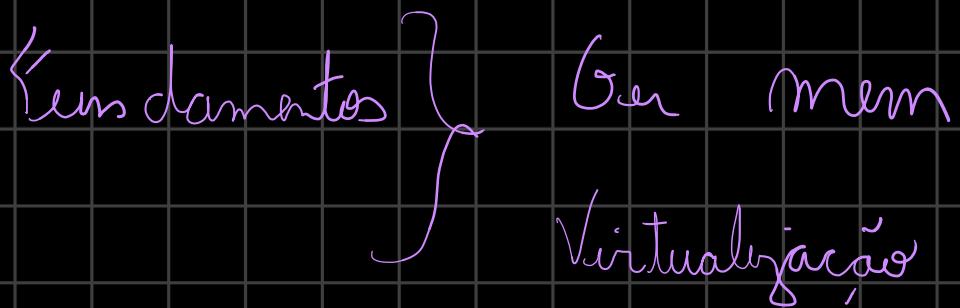
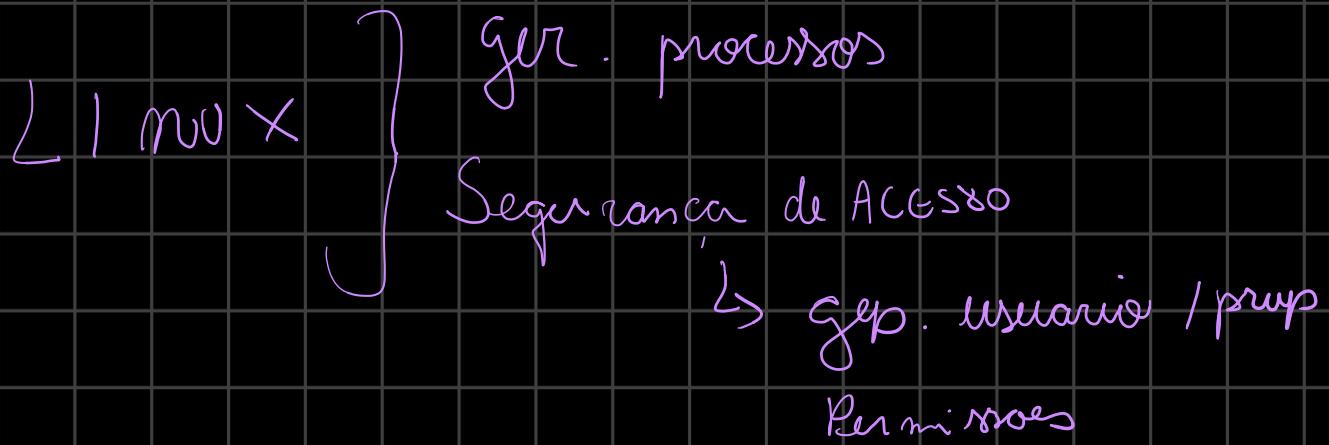


AP2



Processos

- Programa em execução
- 3 Partes Hardware, Software, endereço
- ESTADOS do PROCESSO
 - Pronto : aguardando para usar a CPU
 - EXECUÇÃO : USANDO a CPU
 - ESPERA : AGUARDANDO um Evento

HARDWARE

- EXECUTA O PROCESSO

SOFTWARE

- GERENCIAM CONTEÚDO

- 3 grupos

- identificação
- QUOTAS
- Privilégios

Espaço de Endereço

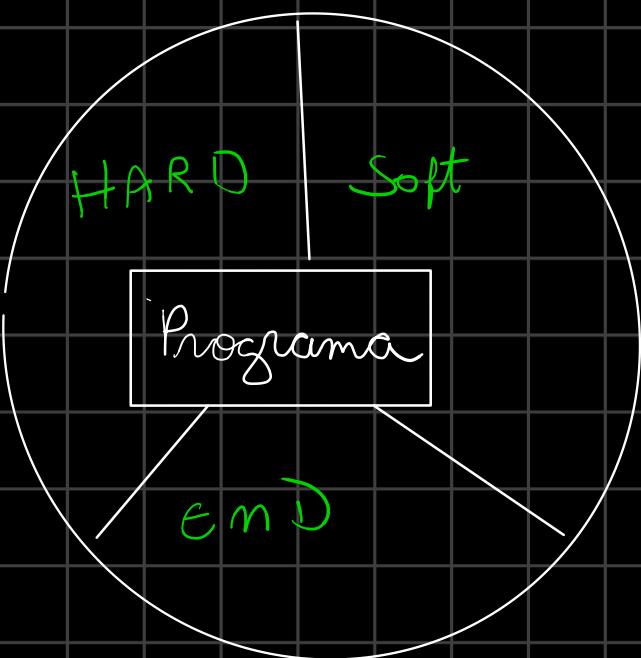
- Armazena os dados e instruções (end de mem)
- "Organizador"

ESTADOS

Pronto \Rightarrow Pronto e aguardar, elo pra usar a CPU

Espera \Rightarrow Esperando evento interno

EXECUÇÃO \Rightarrow USANDO A CPU



ESTADOS

- (A) Pronto \rightarrow EXECUÇÃO (chega a vez de usar a CPU)
- (B) EXECUÇÃO \rightarrow Espera (aguarda recurso externo)
- (C) ESPERA \rightarrow PRONTO (recurso interno fornecido)
- (D) EXECUÇÃO \rightarrow PRONTO (substituir que usa a CPU)

EXECUÇÃO:

Foreground: interage com o usuário

Background: não interage e não exibe a execução no monitor

Tipo de Processos

INTERATIVOS

BATCH \rightarrow Não interage com o usuário, no final apresenta o resultado: "UPLOAD"

DEAMONS \rightarrow PROCESSOS iniciados pelo SO.

(Background)

SLIDES 5

ESTRUTURA DE PROCESSOS

Atributos de um processo

- **PID** → identifica um processo em execução
- **PPID** → identifica o processo pai que gerou a identificação.
- **UID** → identifica o usuário que criou o processo
- **GID** → identifica o grupo

ESTADOS

Pronto: entra na fila

EXECUÇÃO: Usando a CPU

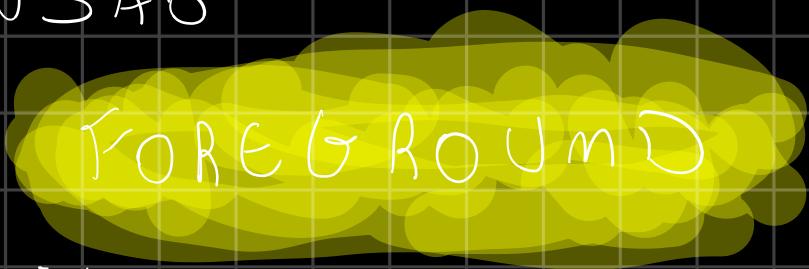
ESPERA: Esperando algum recurso

FINALIZADO: terminado

Zombie: Processo filho é finalizado, e o pai ainda está vivo e não coletou o status de saída do filho.

PAUSA DO: Só o usuário pause mesmo estando na memória no ponto de processamento que parou

EXCUSSÃO



• Pode interagir com o usuário e exibe a execução no monitor

• impedem outros processos de serem iniciados



• Não interage e não exibe sua execução

• não impede outros processos de serem iniciados

COMANDOS

DE
GERENCIAMENTOS

Controle de execução

- & → Executa o processo em 2º plano
- Jobs → Mostra processos parados ou rodado em 2º plano
- fg → Faz um programa no 2º plano ou parado, rodar no 1º plano
- BG → Programa rodar em 2º plano
- PIDOF → Informa o PID de um processo
- CTRLS → São solicitações para os processos.
 - CTRLS + C ⇒ Interrrompe o programa
 - CTRLS + Z ⇒ Pausa o processo, colocando-o 2º plano
 - CTRLS + D ⇒ Envia sinal de fim de arquivo (EOF), causando uma desconexão.
 - CTRLS + L ⇒ Envia sinal para apagar e atualizar a tela
- KILL → Envia sinal para o Processo pelo

PID.

Sintaxe : KILL <-signal> PID

Sinais

- 1 : Desligamento do Terminal (hang up)
- 2 : Equivalente ao CTRL+C
- 3 : sair , equivalente CTRL+D
- 9 : O processo finaliza imediatamente
- 15 : Solicita que o processo finalize normalmente
- 19 : Sinal de parada temporária (CTRL+Z)

ANALISE DE PROCESSOS e Memória

- TOP → Exibe os processos com maior consumo de CPU
- UPTIME → tempo de funcionamento e sua carga
- FREE → Quantidade de memória livre usada
- PS → Informações sobre processos Ativos

Sintaxe: ps [op]

- a : inf de outros usuarios
- U : Exibe o nome de usuário e a hora de inicio do processo
- x : processos não associados ao terminal
- p (pid) : processos cujo número é pid
- o (formato) : especifica campos
 - .CMD - nome do processo | .PCPU - CPU utilization
 - .pid - ID | .PPID - parent ID
 - .UID - user ID | .USER - user name
- pstree → ps em árvore

ALTERAR PRIORIDADE

- nice → Define a prioridade no momento da inicialização

Sintaxe: nice [PR] [comando [argumentos]]

↓

PRI → numero da prioridade, quanto maior o valor menor é a prioridade sendo o padrão (-20 até +29)

. renice → modifica a prioridade. Pode ser aplicada a um processo, usuário ou grupo.

Sintaxe: renice PRI [Op] ID

OP

. -p → id é PID

. -U → id é UID ou user name

. -g → id é GID ou group name