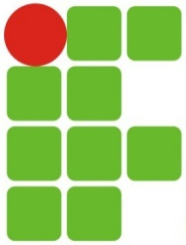

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Instalação e Configuração de Servidores

Linux Server – Gerenciamento de Processos

Prof. Alex Furtunato

alex.furtunato@academico.ifrn.edu.br

Roteiro

- Introdução
- Atributos de um processo
- Sequência de execução
- Classificação de processos
- Controle de tarefas
- Comandos para gerenciamento de processos
- Redirecionamentos e Pipes

Introdução

- Processo é um programa em execução
- Para o SO, um processo é uma estrutura com diversas informações sobre essa execução
- Cada processo tem permissões de acessos e atributos
- Para cada processo em execução, é criado um subdiretório em /proc com informações sobre o processo

Atributos de Processos

- PID – Process ID
- PPID – Parent Process ID
- UID – User ID
- GID – Group ID
- EUID – Effective UID (Setuid)
- EGID – Effective GID

Sequência de Execução

- Inicia no estado de ”ready”
- Escalonador de tarefas decide, de acordo com as prioridades e filas de execução, quando ceder CPU ao processo que passa ao estado de ”running”
- Após sua fatia de tempo terminar, volta ao estado de ”ready”
- Em operações de entrada e saída, enquanto espera fica em estado de ”waiting”
- Processos que terminam sem avisar ao pai, podem ficar em estado de ”zombie”

Classificação de Processos

- Quanto a execução:
  - Foreground
    - Executados através do console
    - Podem interagir com o usuário
    - Exibem as saídas de execução no console
    - Prendem o prompt
  - Background
    - Executados através do console
    - Não interagem com o usuário
    - Não exibem as saídas de execução no console
    - Não prendem o prompt

Classificação de Processos

- Quanto ao tipo:
  - Interativos
    - Iniciados a partir de uma sessão do usuário
    - Iniciam a execução em foreground
    - Entradas e saídas por Stdin, Stdout e Stderr
    - Pode-se utilizar os comandos fg, bg e jobs
  - Em lote (batch)
    - Processos controlados pelos comandos at, batch e cron
    - A saída é enviada por email ao usuário
  - Daemons
    - Processos servidores
    - Normalmente rodam em background

Controle de Tarefas

- Habilidade de suspender e retomar a execução de processos
- O shell associa os processos aos jobs inicializados por ele

```
joao@debian:~$ vi &

[1] 4297
[1]+ Stopped vi
```

Comandos

- ”comando &” - roda comando em background
- bg – Coloca processo em segundo plano
- fg – Coloca processo em primeiro plano
- jobs – Exibe as tarefas em execução
- ps – Exibe informações sobre processos
- top – Exibe processos e consumo de CPU
- kill ou killall – Finaliza processos
- nohup – impede que um processo seja finalizado após fechamento do shell

Alguns Sinais do Sistema

Sinal	Valor	Comentário
HUP	1	Travamento detectado ou finalização do processo controlado
INT	2	Interrupção através do teclado
QUIT	3	Sair através do teclado
ILL	4	Instrução Ilegal
ABRT	6	Sinal de abortar enviado pela função abort
FPE	8	Exceção de ponto Flutuante
KILL	9	Sinal de destruição do processo
TERM	15	Sinal de Término
STOP	17, 19, 23	Interromper processo

Finalizando Processos

- Pode-se usar duas formas para matar processos:
  - kill [opcoes] [sinal] PID
  - killall [opcoes] [sinal] nome_processo

```
joao@debian:~$ kill -9 4297

joao@debian:~$ kill -9 apache2
```

Parando Programas em Execução

- Para parar um programa em execução, utilize <CTRL>+z
  - Com o comando jobs, pode-se ver os programas parados e colocados em segundo plano
  - Utilize o comando fg para retornar o processo para primeiro plano

Rodando Processos em segundo plano

- Normalmente, processos rodando em segundo plano são mortos quando o processo pai é morto
- Para evitar isso, utilize o ”nohup” antes do comando de execução
- Por exemplo, para rodar um prcesso em segundo plano e mantê-lo em execução, mesmo após o logout

```
joao@debian:~$ nohup /sbin/apache2 &
nohup: Ignorando entrada e adicionando saída
em `nohup.out´
[1] 5632
```

Exercício

- Execute o comando: vim
- Pressione CTRL+Z para pará-lo
- Execute o comando: top
- Pressione CTRL+Z para pará-lo
- Use jobs para listas os jobs em execução
- Utilize o comando fg para retornar o jobs 2
- Feche o programa
- Execute fg para retornar o job 1

Redirecionamentos

- Para redirecionar as saídas ou entradas de um comando, utilize:
  - ”comando > alvo” : Redireciona a saida do comando para o alvo que pode ser um arquivo ou um dispositivo
  - ”comando >> alvo” : Redireciona a saida do comando para o alvo que pode ser um arquivo, concatenando ao final do arquivo
  - ”comando < fonte” : Redireciona os dados da fonte como entrada para o comando
  - ”comando << fonte” : Serve, principalmente, para marcar o final de um bloco

Exemplos de Redirecionamentos

```
joao@debian:~$ ls -ax /home > listagem.txt
joao@debian:~$ ls -ax /var >> listagem.txt
joao@debian:~$ cat < listagem.txt
joao@debian:~$ cat << final
> testando
> alooooou alooou
>final
testando
alooooou alooou
```

PIPE

- Envia a saída de um comando para a entrada de outro
- Utiliza o caractere "|" para conectar os comandos

```
joao@debian:~$ ls -ax /home | more
joao@debian:~$ cat < listagem.txt | wc
10 41 245
joao@debian:~$ ps -aux | grep apache2
```