Gerenciamento de pacotes e processos

Processos

Processos são programas em execução, cada processo tem suas próprias permissões e atributos. No Linux cada processo cria um sub-diretórios em **/proc**

PID = Identificação do processo em execução.
PPID= Parent Processo Identification - identifica o processo PAI do processo PID.
UID = Identificação do Usuário iniciou o processo.
GID = Grupo no qual o processo pertence.

No Linux todo PID tem um PPID

Os sistemas operacionais são responsáveis pelo gerenciamento dos processos que é escalonamente, a CPU executa um processo por cada vez, e quando temos mais de um processo em execução, o sistema operacional executa um pouco de cada processo por vez e vai alternando os processos.

Estados de um processo:
Ready - Processo pronto para a execução
Dead - Processo finalizado ou morto
Zombie - Processo que o processo Pai (PPID) foi finalizado, por tanto o PID não consegue mais executar e será finalizado.
Running - em execução
Waiting - Em espera, para ser executado
Interrompido - temporariamente parado

Ver processo que estão rodando a partir desse terminal e o PPID de cada um deles
$ ps -f

Ver processo específico
$ ps -f número-ID

Flag
-f # Ver informações de processos formatadas
- a # Ver processos de todos os usuários
-u # Ver nome do usuário e a hora de inicio
- el # Ver processos em detalhe
Legenda:
S = Sleeping (sem atividade)
R= Running (executando)
D= waiting (aguardando um dispositivo de I,/O)
T = gestopt (suspenso, parado)
Z= Zombie (órfão)

Exibe processo ativos em forma de árvore
$ pstree -flag # Ver todos os processo

$ pstree -flag PID # Ver a partir desse PID

Flag:
-a mostrar comandos
-c não compactar sub-árvore
-p mostra PID de cada processo
-l mostrar detalhes

Mostrar consumo de CPU e memoria (similar a gerenciador do windows)
$ top # Todos os usuários

$ top [flag] nome-usuário # Ver usuários específicos

Flag:
-u # Verificar processo de usuário específico
- U # Não mostrar esse usuário

Exibir as informações de uso da memória em megabytes
$ free -m

Exibir o tempo de funcionamento do sistema e sua carga
$ uptime

Tipos de processos:
- Processos Interativos= Dependem da interação com o usuário
- Processo em lote (batch, at e cron)= processos agendados, não são interativos.
- Daemon= processos ou serviços que permanecem em execução enquanto o sistema estiver funcionamento. Processo não interativo (não interativo).

Tipos de execução de processos:

- **Foreground ou primeiro plano**: execuções interativa, necessitam da interação direta do usuário, e "prendem" o shell, ou seja não conseguimos inicializar outro processo no shell, enquanto estão sendo executados.
- **Background ou segundo plano**: execuções não são interativas e não prendem o shell, são inicializados no terminal de comando, mas não necessitam de interação com o usuário, dessa forma podemos utilizar o shell para inicializar outros comandos.

Criação do Processo
- **fork** Cria um novo processo filho, executa, mata processo filho e volta ao processo pai. **Esse é o padrão do BASH ou Shell do Linux.**

Exemplo:

$ cat /etc/group/
$ ls /home/usuário

- **exec** O novo comando toma controle do processo atual, ouseja, o comando novo substituir o processo anterior. Basicamente executa o novo comando e mata o anterior. Por tanto com o novo comando após ser executado, fecha o shell

Exemplo:
$ exec cat /etc/group/
$ exec ls /home/usuário

Passar comando para background

- Ctrl+Z # Parasila o job

- Adicionar o & após o comando

Exemplos:
$ nano texto.txt & # abri texto.txt
$ sleep 60 & # bloquear teclado por 60 sequndos

Passar comando do background para foreground
$ fg % PID

Ver Jobs
O comando **jobs** mostra os processos filhos do BASH e que estão em background, informando o estado de cada um deles, por exemplo se estão em execução, feito ou interrompido.

$ jobs

O comando **kill** é utilizado para sinalizar alterações para um processo, inclusive encerrar ou matar um processo.
$ kill -9 PID # O processo deve ser encerrado imediatamente
$ kill -15 PID # O processo deve ser encerrado após realizar as atividades necessárias.
$ kill -20 PID # processo deve ser parado imediatamente
$ kill -18 PID # processo deve continuar sua execução

Outra maneira de encerra o processo imediatamente é: **Crtl +C**

Prioridades
A prioridade varia entre **-20** (maior prioridade) e **19** (menor prioridade), no entanto, apenas o usuário root pode atribuir prioridades de valores negativos. O comando **nice** é para executar um processo com a prioridade diferente da padrão, enquanto que o comando **renice** modifica a prioridade de um processo em execução e pode ser aplicado a um processo, usuário ou grupo.

$ nice -flags prioridade comando-a-ser-executado

$ renice prioridade -flags nome-usuario-ou-grupo-ou-numero-processo

Flags:
-p processo
-u usuário
-g grupo

Pacotes

Instalação, listagem, atualização e desinstalação de programas usando a distribuição Debian. O comando **dpkg** é usado para instalar programas que o arquivo é baixado da internet e o arquivo compactado está no computador. O **apt** é usado baixar diretamente de repositórios na internet, resolve automaticamente as dependências, instala os programas requeridos para a instalação de determinado pacote

Ver os repositórios que o apt irá fazer as buscas no arquivo
/etc/apt/source.list

Comandos:
$ dpkg -flags nome-programa

- i # Instalar o programa (se houver algum dependência o programa não funcionará direito).man
-P # Desinstalar pacote.
--configure

$ dpkg -l # Listar os programas instalados.

Atualizar os pacotes
$ apt-get install nome-pacote

$ apt-get install wget

$ apt-cache search clock