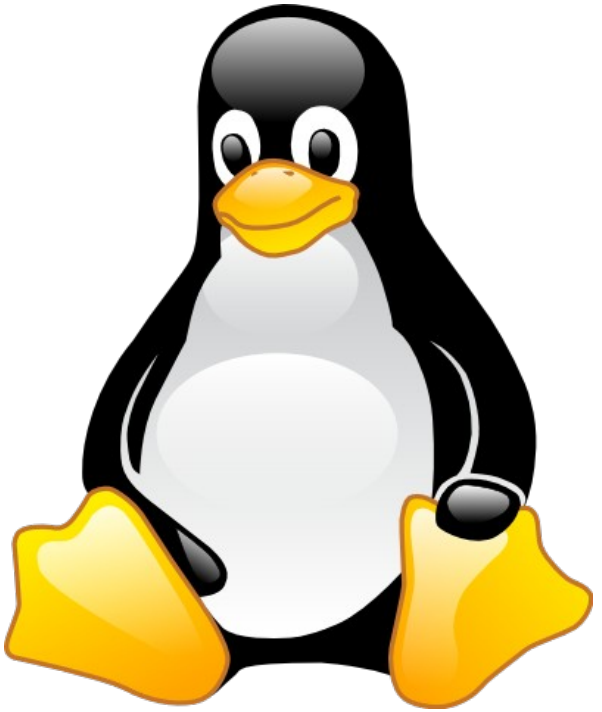
Linux System Administration 455

Aula 12 – 455



Aula 12 – 455

ls /proc

Nesse diretório, você verá vários arquivos com informações gerais que o kernel nos fornece, por exemplo os arquivos:
cpuinfo, ioports, meminfo, interrupts, mounts, swaps e muitos outros.

Vale a pena ver o conteúdo desses arquivos para aprender mais sobre eles.

No /proc, você pode entrar em um número (diretório) que identifica um processo e listar o conteúdo dele, exemplo:

cd /proc/3512

cat cmdline

/usr/lib/iceweasel/firefox-bin-afirefox

Aula 12 – 455

Dois arquivos interessantes de explorar nesses diretórios dos processos são:

cmdline

status

Para saber mais sobre o /proc:

man proc

Aula 12 – 455

ps aux | more

Explicando o comando:

ps aux

a = todos os processos

u = de todos os usuários

x = e até processos sem controle pelo terminal

Aula 12 – 455

USER - Usuário responsável pelo processo.
PID - Número que identifica o processo, não se repete!
% CPU - Quanto de CPU ele usa!
% MEM - Quanto de MEM ele usa!
VSZ - Tamanho virtual do processo;
RSS - Indica a quantidade de memória usada (em KB);
TTY - Terminal que gerou o processo.
? - sem terminal
STAT - Estado do processo, sendo representado por uma letra.
R - executável;
D - em espera no disco;
S - Suspenso;
T - interrompido;
Z - Zumbi.
COMMAND - Nome do processo, ou seja, o comando em si.

Aula 12 – 455

Com a opção f em ps, posso ver em forma de árvore, caso o processo tem processos filhos:
ps faux | more

Uma alternativa ao ps faux é:
pstree

E se quisermos monitorar a execução dos processos vendo uma tabela que se atualiza de tempos em tempo? Tem algum comando que faça isso?
top
man top

Aula 12 – 455

Para matar um processo, você tem duas formas:

Ou pelo PID (número do processo) (2ª coluna)

Ou pelo COMMAND (nome do processo) (última coluna)

Lembrando que você precisa ser o dono do processo ou o usuário root para destruí-lo.

Então podemos visualizá-lo dessa maneira:

```
# ps aux | awk '{print $2,$11}'
```

Onde estarei mostrando apenas o PID e o nome dos processos!

Aula 12 – 455

Aí, é só pegar o PID dado pelo comando ps e matar o processo:

kill -9 PID

-9 - estou forçando para que o mesmo seja morto.

Aula 12 – 455

Sintaxe do comando kill:

kill SINAL PID

Se o sinal for omitido é usado 15 como padrão.

Aula 12 – 455

Ou ainda, pegar o nome e usar o comando:

killall firefox

Onde:

killall - comando para matar um processo pelo nome.

Na prática no comando kill apenas usamos o -9 e o -15 para matar um processo que esteja travado ou atrapalhando o bom funcionamento do sistema!

Aula 12 – 455

Missão

Pesquisem acerca dos possíveis sinais que podemos enviar para os processos e o que cada um deles significa.

2 minutos!

Aula 12 – 455

Resposta:

Mas como podemos conhecer os demais sinais que podemos usar nos processos?

kill -l

Para descobrir para que serve cada sinal desse:

man 7 signal

Aula 12 – 455

Alguns sinais mais usados:

STOP - esse sinal tem a função de interromper a execução de um processo e só reativá-lo após o recebimento do sinal CONT;
CONT- esse sinal tem a função de instruir a execução de um processo após este ter sido interrompido;
TERM - esse sinal tem a função de terminar completamente o processo, ou seja, este deixa de existir após a finalização;
KILL - esse sinal tem a função de "matar" um processo e é usado em momentos de criticos.
HUP - pode ser usado para que um serviço leia novamente seu arquivo de configuração.

Aula 12 – 455

Para localizarmos esse processo, usamos um comando muito legal:

pgrep firefox

Esse comando irá me retornar qual o número PID que está o cron.

Será que ainda tem mais algum comando para que eu possa obter o PID do Firefox, por exemplo?

pidof firefox

Aula 12 – 455

Então com o número em mãos, já posso matar esse processo:

kill -15 PID

Ou

kill -s SIGTERM 3222

Ou ainda:

pkill -15 firefox

Aula 12 – 455

Para continuar usando nosso shell vamos colocar o comando em segundo plano:

$ cp -R dir dir2 & (Apenas exemplo)

Onde o comando para colocar em background é apenas o & no final do comando em si.

Para ver o processo que está rodando em background:

#jobs

Onde o comando jobs irá me mostrar o processo que está em background.

Aula 12 – 455

Caso eu queira colocar esse processo novamente para primeiro plano eu pego o número que o jobs me passou e digito:

#fg 1

Onde fg (foreground) eu trago o processo para primeiro plano.

Mas porque usei o número 1 junto ao comando?

Porque este foi o número do job relacionado ao processo que eu quero trazer para foreground.

Exemplificando na prática:

#updatedb &

[1] 14082

Que é o comando que atualiza a base de dados do comando locate.

Aula 12 – 455

Colocou meu processo em segundo plano. E eu continuo mexendo no terminal.

#jobs

[1]+ Running updatedb &

Mostra o processo que está em segundo plano e o status dele (Rodando)

fg 1

updatedb

Aula 12 – 455

Dica:

Para terminar a execução de um programa em background posso matá-lo pelo comando kill, pois ele aparece no ps aux.

Ou trago para primeiro plano e dou um CTRL+C, que é um Sinal de Interrupção!

Aula 12 – 455

Então, para fazer isso temos que primeiro dar uma parada no processo:

CTRL+Z (No shell que está travado com o processo)

Essa seqüência de tecla PÁRA (STOP) o processo, mas o mesmo fica em segundo plano parado.

Então damos jobs novamente:

#jobs

E veremos que ele está lá, só que agora parado!!! Pegamos o número dele que o jobs me retornou e falamos para ele continuar em segundo plano agora.

bg N°

Aula 12 – 455

Na prática:

Abro um arquivo no VI

$ vi teste.txt

Quero usar o meu terminal, então quero parar o Vi por um tempo.

Aula 12 – 455

CTRL+Z
[1]+ Stopped vi teste.txt

Pronto, está parado em segundo plano!

jobs
[1]+ Stopped vi teste.txt

E agora como faço para trazer o vi novamente para o meu shell?
#fg 1

Aula 12 – 455

O nice é usado para quando quero iniciar um processo com uma determinada prioridade.

Já o renice é para mudar a prioridade de um processo que já esteja rodando .

Mas os dois definem prioridades tanto baixa quanto alta.

Esses comandos costumam cair na LPI!

Lembrando que:

-20 ----------10---------- 19

+ alta----------padrão---------- + baixa

Aula 12 – 455

Vou iniciar um processo com prioridade alta:

cd /

nice -n -19 find /boot -name menu.lst

Aula 12 – 455

Para ver a prioridade, vá em outro terminal e digite:

ps -lax

Onde a opção -l do comando ps é para listar em formato longo, ou seja, todas as informações!

Aula 12 – 455

Agora para mudar a prioridade de um processo que está em execução, preciso do PID dele!
ps aux

Altere a prioridade de um determinado processo:
#renice 19 -p PID

Aula 12 – 455

Sua sintaxe é:

fuser opção caminho arquivo/diretório

-k - finaliza o processo que utiliza o arquivo/diretório;

-v - o resultado é mostrado em um padrão de exibição semelhante ao comando ps.

Aula 12 – 455

Então, posso usar o comando abaixo para verificar quem está usando o cd-rom:

fuser -v /media/cdrom

USER PID ACCESS COMMAND

/media/cdrom: leo 5125 ..c.. bash

Aula 12 – 455

Se você quiser ver somente quem é o usuário:

fuser -u /media/cdrom

/media/cdrom: 5125c(leo)

Aula 12 – 455

fuser -k /media/cdrom

Tente ver algo agora:
fuser -v /media/cdrom

Outro exemplo é verificar o PID de um serviço de rede, como por exemplo o SSH:

fuser -v 22/tcp

USER PID ACCESS COMMAND
22/tcp: root 3025 F.... sshd

Aula 12 – 455

O comando nohup permite que o processo fique ativo mesmo quando o usuário faz logout.

Em sistemas baseados em Unix (Unix-Like) é normal que eles interrompssam processos caso seu dono não esteja mais ativo, por isso, o nohup pode ser muito útil para manter o processo ativo mesmo se o dono fizer logout.

Sua sintaxe é:

nohup comando