

Sistema Operacional

Unidade 7.2 – Gerenciamento de Processos

Ubuntu 12.04 LTS
Precise Pangolin



QI ESCOLAS E FACULDADES
Curso Técnico em Informática

SUMÁRIO

INTRODUÇÃO	3
PS.....	3
Atributos de ps	3
Exemplos de ps.....	3
TOP	4
Atributos	5
KILL	5
Exemplos do comando <i>kill</i>	5
KILLALL	6
Exemplo.....	6
XKILL	6
BIBLIOGRAFIA.....	6

INTRODUÇÃO

Um processo é basicamente um programa em execução. Quando clicamos em um programa para abri-lo (por exemplo, a calculadora) ele é carregado para a memória e se transforma em um processo em execução. Como utilizamos muitos programas ao mesmo tempo, precisamos de mecanismos para controlar estes processos.

Um processo possui um proprietário (usuário), um estado (em espera, em execução, etc.), prioridade, utilização de memória.

No Linux podemos gerenciar os processos de forma bastante simples, via modo texto. Nesta unidade aprenderemos os comandos necessários para manipular estes processos.

PS

O **ps** (*process*) é um comando muito importante nas distribuições Linux. Ele é responsável por exibir todos os processos que estão em execução atualmente no sistema operacional, bem como fornecer informações importantes sobre eles.

ATENÇÃO: Se digitarmos apenas “ps” no terminal, ele informará só os processos corrente do usuário logado. Para abrir todos os processos que estão em execução no sistema, precisamos utilizar os atributos do comando.

Atributos de ps

-a	Mostra todos os processos existentes
-f	Exibe a árvore dos processos
-u	Exibe o nome do usuário que abriu o processo e hora que foi aberto
-x	Exibe os processos que não estão associados a terminais

Exemplos de ps

Mostra todos os processos ocorrendo no sistema, bem como quem abriu (usuário), quantidade de memória e processamento que está sendo utilizado, hora, número de identificação do processo(PID), entre outros.

```
ps -aux
```

```
ps -lax
```

Essa combinação de atributos mostrará mais informações sobre os processos em execução no sistema, são elas:

User	Nome do usuário dono do processo
Uid	Número de identificação do usuário dono do processo
Pid	Número de identificação do processo
Ppid	Número de identificação do processo pai;
%cpu	Porcentagem do processamento usado;
%mem	Porcentagem da memória usada
Vsz	Indica o tamanho virtual do processo
Rss	Sigla de R esident S et S ize, indica a quantidade de memória usada (em KB);
Tty	Indica o identificador do terminal do processo
Start	Hora em que o processo foi iniciado;
Command	Nome do comando que executa aquele processo;
Pri	Valor da prioridade do processo;
Ni	Valor preciso da prioridade (geralmente igual aos valores de PRI);
Wchan	Mostra a função do <i>kernel</i> onde o processo se encontra em modo suspenso;
Stat	Indica o estado atual do processo, sendo representado por uma letra: R - executável; D - em espera no disco; S - Suspenso; T - interrompido; Z - Zumbi. Essas letras podem ser combinadas e ainda acrescidas de: W - processo paginado em disco; < - processo com prioridade maior que o convencional; N - processo com prioridade menor que o convencional; L - processo com alguns recursos bloqueados no <i>kernel</i> .

TOP

O comando *top* verifica os processos, porém por padrão ele atualiza a lista de processos de 3 em 3 segundos.

Atributos

-d	Indicar o período de atualização em segundos.
-i	Faz o <i>top</i> ignorar processos no estado zumbi;
-s	Executa o <i>top</i> em modo seguro.

Obs: Apertando a barra de espaço o comando *top* atualiza a lista de processos automaticamente.

Vejamos um exemplo.

Mostra os processos em execução, porém atualiza a lista de 30 em 30 segundos.

```
top -d 30
```

ATENÇÃO: Para encerrar a exibição do comando *top* é só apertar a tecla “q”.

KILL

O comando *kill* envia um sinal para um processo travado, isso através de seu número de identificação, ou seja, através de sua “PID”. PID é a identificação do processo. Todo processo é identificado através de um número denominado PID.

Se por acaso o processo não encerrar só digitando *kill* seguido do número da PID, podemos encerrar o processo a “força” utilizando o sinal “-9”.

Exemplos do comando *kill*

Encerra um processo a força. O sinal -9 é chamado de SIGKILL, ou seja, sinal para matar.

```
kill -9 4320
```

Encerra um processo com a PID 4320.

```
kill 4320
```

Para visualizar a lista de sinais que podem ser enviados aos processos, digite:

```
kill -l
```

O atributo l mostra a lista de sinais disponíveis.

KILLALL

Esse comando encerra uma árvore de processos. Todo processo tem um número que o identifica, porém o nome dele é o mesmo. Isso significa que se você abrir, por exemplo, a calculadora três vezes, cada uma receberá um número de *PID*, porém todas terão o mesmo nome: *gcalctool*.

Podemos matar cada instância através do comando *kill*, ou encerrar todas ao mesmo tempo matando através do comando *killall*.

O comando killall encerra processos através de seu nome.

Exemplo

Encerra todas as instancias de gcalctool abertas.

```
killall gcalctool
```

XKILL

O *xkill* é uma maneira divertida de encerrar processos. Basta abrir o terminal e digitar *xkill*, após apertar a tecla *enter* o ponteiro do mouse se transformará em um “x”. Com esse “x” podemos clicar em qualquer programa travado que ele o encerrará na mesma hora.

ATENÇÃO: O comando *xkill* só funciona se for executado a partir de um terminal aberto no modo gráfico.

BIBLIOGRAFIA

SILVA, Gleydson Mazioli. **Guia Foca GNU/Linux**. Disponível em <http://www.guiafoca.org/>. 2010.

SEGURANÇA DIGITAL. **Kill, killall e xkill**. Disponível em <http://www.segurancadigital.info/dicas/60-linux-geral/417-kill-killall-e-xkill>. Acesso em agosto de 2012.

ALECRIM, Emerson. **Processos no Linux**. Disponível em <http://www.infowester.com/linprocessos.php>. Acesso em agosto de 2012.