Gerência de Processos

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

20 de Setembro de 2019

O comando *ps*

```
$ ps aux
```

Exibe informações detalhadas dos processos em execução no sistema. Se quiser saber se um script com nome *meuscript.sh* está em execução?

```
$ ps aux | grep meuscript
```

Se quero ver todos os processos do meu usuário?

```
$ ps aux | grep usuario
```

Detalhe: em ambos os casos acima, uma linha extra é retornada para o próprio processo que executa o *grep*. Para matar um processo, existe o comando *kill*.

```
$ kill PID
```

O PID é a segunda coluna da saída do *ps aux*.

O uso do &

Caso você deseje executar um comando e ter de volta o terminal?

```
$ ./meuscript.sh [parametros] [redirecionamentos] &
```

- Você, a partir do seu script, iniciar processos da mesma forma.
- A variável $! armazena o PID do último processo criado.
- Se você terminar a sessão, ou seu script finalizar, o que acontece com os processos criados depende da distribuição. Eles podem ser finalizados ou continuar.
- Para garantir que eles continuam a executar, use o *nohup*.

```
$ nohup ./meuscript.sh &
```

A saída do script será direcionada para *nohup.out* e você pode ter certeza que ele continuará executando até ser finalizado pelo *kill*.

O comando *screen*

O & e o *nohup* são bons para criar *daemons*.
Mas se você quiser apenas manter uma sessão ativa enquanto entra e sai na máquina, o *screen* é uma boa opção.

```
$ screen
```

Você agora pode deixar qualquer comando executando. Para *desanexar* a tela, digite crtl + a + d.
Você pode fazer *logout* e um novo *login*. Para recuperar a sessão, digite:

```
$ screen -r
```

Mais opções do *screen*

- Criar um terminal em uma tela: ctrl + a + c
- Trocar entre as telas: ctrl + a + tab
- Ir para a próxima tela: ctrl + a + espaço
- Ir para a tela anterior: ctrl + a + backspace
- Ir para a n-ésima tela: ctrl + a + [n]
- Listar as telas abertas: ctrl + a + “
- Dividir a tela na vertical: ctrl + a + |
- Dividir a tela na horizontal: ctrl + a + S
- Encerrar a divisão de telas: ctrl + a + Q

O comando *tmux*

É similar ao *screen* porém mais recente.
Inicializando uma sessão:

```
$ tmux
```

Podemos também nomear as sessões:

```
$ tmux new -s programando
```

Listando sessões ativas:

```
$ tmux ls
```

Retornar a uma sessão específica:

```
$ tmux attach -t 0
```

Mais opções do *tmux*

- Criar um novo painel: CTRL + b + c
- Dividir um painel: CTRL + b + %
- Alternar entre painéis: CTRL + b + *numero*
- Fechar um painel: CTRL + b + d
- Voltar ao painel anterior: CTRL + b + p
- Avançar ao próximo painel: CTRL + b + n
- Desanexar da sessão: CTRL + d

Exercício em Sala

O campo VZS da saída do **ps aux** equivale a quantidade de memória virtual em *kilobytes* que o processo utiliza.
Vamos fazer um *script* chamado *consumoDeMemoria.sh* que receba como parâmetro o nome do usuário e imprima quanto o mesmo está consumindo da memória virtual.

Exercício em Sala

Vamos fazer um *script* chamado *IPsAtivos.sh* que receba como parâmetros os três primeiros octetos de uma sub-rede /24 e verifique quais IPs da rede estão ativos.
Exemplo:
$ ./IPsAtivos.sh 192.168.0.
192.168.0.1 on
192.168.0.54 on
192.168.0.101 on
Esse *script* pode demorar muito. Use algumas das técnicas apresentadas hoje para guardar o resultado em algum lugar e depois verificá-lo.

Atividade - Parâmetros

Crie o diretório *atividades/atividade07*. Escreva um *script* chamado *alertaDiretorio.sh* que recebe como **parâmetros** um valor inteiro que serve como intervalo de tempo em segundos e um nome que indica o caminho de um diretório.

Atividade - Ações

A cada intervalo, a quantidade de arquivos em um diretório será analisada. Caso a quantidade de arquivos se altere entre duas verificações, o *script* deve atualizar um arquivo chamado *dirSensors.log* com as seguintes informações:

- A data que a alteração foi percebida.
- Quantos arquivos haviam.
- Quantos existem agora.
- Quais foram alterados, adicionados ou removidos.

Na mesma pasta da atividade, crie um diretório chamado *diretorioMonitorado* para testar.
Deixe seu *script* executando em uma sessão *screen* ou *tmux*, monitorando o diretório a cada 5 segundos.

Atividade - Formato do Arquivo

```
$ ./alertaDiretorio.sh 5 diretorioMonitorado
[25-09-2018 12:59:51] Alteração! 3->2. Removidos: notas.txt
[25-09-2018 13:04:51] Alteração! 2->4. Adicionados: a.txt, b.txt
[25-09-2018 13:09:51] Alteração! 4->3. Removidos: a.txt
[25-09-2018 13:14:51] Alteração! 3->2. Removidos: b.txt
```