

Sessão 3 — Processos



As atividades desta sessão serão realizadas na máquina virtual *Client_Linux*.

1) Descobrindo o número de processos em execução

1. Quantos processos estão sendo executados na máquina no momento? Use o comando `wc` para contá-los.
2. Faça um *script* que liste o número de processo que cada usuário está executando.

2) Descobrindo o PID e o PPID de um processo

1. Quais os valores de `PID` e `PPID` do shell que você está utilizando no sistema?
2. Faça um *script* que liste todos os processos que foram iniciados pelo processo `init`. A lista não deve conter mais de uma ocorrência do mesmo processo.

3) Estados dos processos

1. Qual o status mais frequente dos processos que estão sendo executados no sistema? Você saberia explicar por quê?

4) Alternando a execução de processos

1. Execute o comando `$ sleep 1000` diretamente do terminal.
2. Pare o processo e mantenha-o em memória.
3. Liste os processos parados.
4. Coloque-o em *background*.
5. Verifique se o comando `sleep 1000` está rodando.
6. É possível cancelar a execução desse comando quando ele está rodando em *background*? Caso seja possível, faça-o.

5) Identificando o RUID e o EUID de um processo

1. Logado como o usuário `aluno`, execute o comando `passwd` no seu terminal. Antes de mudar a senha, abra uma segunda console e autentique-se como `root`. Verifique o `RUID` e o `EUID` associados ao processo `passwd`. Esses valores são iguais ou diferentes? Você saberia explicar por quê? Por fim, cancele a execução do processo `passwd`.

6) Definindo a prioridade de processos

1. Verifique as opções do comando `nice` e em seguida, execute o comando abaixo, verificando sua prioridade, utilizando o comando `ps`:

```
# nice -n -15 sleep 1000 &
[1] 2289
```

2. Repita o comando do primeiro item, passando para o comando **nice** o parâmetro **-n -5**. Verifique como isso afeta a prioridade do processo. Ela aumentou, diminuiu ou permaneceu a mesma?

7) Editando arquivos crontab para o agendamento de tarefas

Neste exercício, trabalharemos com o comando **crontab**, utilizado para editar os arquivos **cron** do agendador de tarefas do sistema. Esses arquivos serão verificados pelo **daemon cron** periodicamente em busca de tarefas para serem executadas pelo sistema.



Para entender o funcionamento do **crontab**, o primeiro passo é ler as páginas do manual relevantes. Para o comando **crontab** em si, consulte a seção 1 do manual:

```
$ man 1 crontab
```

Para o formato de um arquivo de configuração **crontab**, consulte a seção 5:

```
$ man 5 crontab
```

1. Existe alguma entrada de **crontab** para o seu usuário?
2. Que opção deve ser usada para editar o seu arquivo de **crontab**?

8) Agendando uma tarefa no daemon cron

Neste exercício, será necessário enviar mensagens de correio eletrônico. Para isso, você deverá utilizar o comando **mail**; o instrutor pode fornecer as informações básicas sobre ele. Um exemplo do uso desse comando para enviar uma mensagem ao endereço **fulano@dominio** com o assunto *Mensagem de teste* é:

```
$ mail fulano@dominio -s "Mensagem de teste" < /dev/null
```

1. Configure o **crontab** para que uma mensagem de correio eletrônico seja enviada automaticamente pelo sistema, sem interferência do administrador às 20:30 horas.
2. Como verificar se a configuração foi feita corretamente?
3. Qual o requisito fundamental para garantir que a ação programada será executada?
4. Há como confirmar se a mensagem foi efetivamente enviada, sem consultar o destinatário?
5. Dê dois exemplos de utilização desse mecanismo para apoiar atividades do administrador de

sistemas.

6. Faça um script que liste os arquivos sem dono do sistema e envie a lista por e-mail ao usuário root.
7. Agende no crontab do usuário **root** o script do item 6, de modo que ele seja executado de segunda a sexta às 22:30 horas.

9) Listando e removendo arquivos crontab

1. Liste o conteúdo do seu arquivo de **crontab** e, em seguida, remova-o. Quais as opções utilizadas para executar as ações demandadas?

10) Entendendo o comando exec

1. Execute o comando **\$ exec ls -l**. Explique o que aconteceu.