

Vitor Hugo Klein

RA:2577895

## Atividades Guiadas e Elaboradas para Comandos de Processos

### Atividade 1: Simulação de Monitoramento de Recursos com Processos

- **Objetivo:** Criar e monitorar múltiplos processos para simular um cenário onde o uso de recursos do sistema precisa ser diagnosticado.
- **Passo a Passo:**
  1. Execute múltiplos comandos em background:  

```
ping google.com 64 > ping_test_um.log &  
ping bing.com 64 > ping_test_dois.log & ping  
yahoo.com 64 > ping_test_tres.log &
```
  2. Liste os processos em execução usando: `ps ax`  
  
Identifique os PIDs dos três processos ping.
  3. Priorize o processo associado ao ping `yahoo.com`: `renice -n -5 -p <PID_yahoo>`
  4. Monitore os processos em tempo real (se `top` estiver disponível) ou liste novamente com: `ps -axl`
  5. Finalize todos os processos ping utilizando: `kill <PID1> <PID2> <PID3>`
- **Resultado Esperado:** Alunos devem entender como criar, monitorar, priorizar e finalizar processos em um ambiente simulado.

---

### Atividade 2: Criando um Script para Gerenciamento de Processos

- **Objetivo:** Criar um script que automatize a execução de comandos, monitore processos e exiba relatórios.
- **Passo a Passo:**
  1. Crie um arquivo chamado `process_monitor.sh`  
  
`nano process_monitor.sh`

2. Insira o seguinte código no script:

```
#!/bin/sh
echo "Iniciando processos sleep em background..." sleep 100
&
echo "Processo 1 iniciado com PID: $!" sleep
150 &
echo "Processo 2 iniciado com PID: $!" sleep
200 &
echo "Processo 3 iniciado com PID: $!"

echo "Listando processos em execução:" ps
ax | grep sleep

echo "Finalizando todos os processos de sleep..." killall sleep
echo "Todos os processos foram finalizados!"
```

3. Salve e saia do editor.
4. Torne o script executável:

```
chmod +x process_monitor.sh
```

5. Execute o script:

```
./process_monitor.sh
```

- **Resultado Esperado:** Alunos criarão um script básico que inicia processos, monitora e finaliza automaticamente.

---

### Atividade 3: Estudo da Hierarquia de Processos

- **Objetivo:** Identificar a relação pai-filho entre processos, manipulá-los e analisar seu comportamento.
- **Passo a Passo:**
  1. No terminal, execute o seguinte comando para iniciar um shell filho:

```
s
h
```

2. Dentro do novo shell, inicie um processo sleep em background:

```
sleep 300  
&
```

3. No shell pai, liste os processos para encontrar o PID do novo shell e do processo sleep

```
ps axf
```

4. Finalize o shell filho:

```
kill <PID_shell_filho>
```

5. Liste novamente os processos:

```
ps  
ax
```

Observe o que aconteceu com o processo sleep

- **Resultado Esperado:** Alunos devem perceber que o processo órfão foi adotado pelo processo init
-