



Atividade - Aula 29/07

Atenção: Vale ressaltar que esta atividade será usada como critério para a contabilização de sua frequência de aula.

Prazo de Entrega: 02/08/2021

Aluno: GUILHERME LUCAS PEREIRA BERNARDO

1. Conceitue processo.

R: Os processos são as chaves de um sistema operacional, mantendo a capacidade das operações pseudo concorrentes, eles contém os códigos dos programas que estão sendo executados e suas atividades atuais

2. Quais são os quatro eventos principais que fazem os processos serem criados? Explique cada um.

R: São 4 eventos, o início do sistema que é onde os processos em primeiro plano interagem com o usuário e os processos em segundo plano são Daemons; Uma chamada de sistema que basicamente é um processo fazendo requisição do sistema para criar outros processos para auxiliar sua execução; O usuário fazer uma chamada de sistema que é quando o usuário executa uma nova aplicação ou comando, clicar em ícones de executáveis, etc; Por fim temos quando o usuário desencadeia tarefas em lote, podendo ser submetido mesmo que remotamente as propriamente ditas tarefas em lote.

3. O que são *threads*? Exemplifique com um exemplo o funcionamento de *threads* dentro de um processo.

R: são os microprocessos dentro de um processo, executados paralelamente a outras threads. um exemplo de threads dentro de um processo seria um jogo, que pode ser modelado com linhas de execução diferentes, mas executando ao mesmo tempo e um único processo, como uma linha execução destinada ao desenho da imagem na tela e a outra para o tratamento do áudio do jogo

4. Execute as seguintes instruções em um sistema operacional Linux e apresente uma Captura de Tela (screenshot) quando solicitado nas instruções.

- Abra o terminal (prompt de comando)
- Execute o comando abaixo que irá dizer ao comando ps para mostrar todos os processos em execução:
 - `$ ps -ef`
- Execute o comando abaixo que irá apresentar os processos em formato de uma árvore:
 - `$ pstree`
- Execute o comando abaixo que irá criar um processo de espera por 10 minutos:
 - `$ sleep 10m &`

- Execute o comando abaixo que irá apresentar informações sobre o processo sleep, executado no comando anterior:
 - \$ ps -aux | grep "sleep"
- Execute o comando abaixo que irá identificar o PID (identificador do processo) no SO:
 - \$ pidof sleep
- Execute o comando abaixo usando o número apresentado no comando anterior para finalizar antes do tempo o processo sleep: ◦ \$ kill PID
 - # exemplo kill 12555 < este foi obtido com pidof
- Apresente a captura de tela do comando anterior.

R:

```

Ubuntu [Executando] - Oracle VM VirtualBox
Atividades Terminal 31 de jul 15:40
guilherme@guilherme-VirtualBox: ~/Área de Trabalho

systemd-journal
systemd-logind
systemd-resolve
systemd-timesyn---(systemd-timesyn)
systemd-udev
udisksd---4*[{udisksd}]
unattended-upgr---(unattended-upgr)
upowerd---2*[{upowerd}]
whoopsie---2*[{whoopsie}]
wpa_supplicant

guilherme@guilherme-VirtualBox:~/Área de Trabalho$ sleep 10m &
[1] 3171
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ ps -aux | grep "sleep"
guilher+ 3171 0.0 0.0 16716 580 pts/1 S 15:36 0:00 sleep 10m
guilher+ 3173 0.0 0.0 17560 724 pts/1 S+ 15:36 0:00 grep --color
=auto sleep
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ pidof sleep
3171
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ kill PID # exemplo 3171
bash: kill: PID: argumentos devem ser IDs de trabalhos ou processo
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ kill PID 3171
bash: kill: PID: argumentos devem ser IDs de trabalhos ou processo
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ kill PID
bash: kill: PID: argumentos devem ser IDs de trabalhos ou processo
[1]+ Terminado sleep 10m
guilherme@guilherme-VirtualBox:~/Área de Trabalho$ kill PID 3171
bash: kill: PID: argumentos devem ser IDs de trabalhos ou processo
b Mostar aplicativos ) - Processo inexistente
guilherme@guilherme-VirtualBox:~/Área de Trabalho$
  
```

