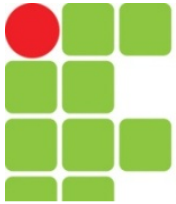

Instituto Federal de Santa Catarina – IFSC
Campus São José

FIC Linux
Processos

Prof. Francisco de Assis S. Santos, Dr.

São José, 2015.

Pacote BrOficce/LibreOffice

- **Writer: Editor de textos**
- **Calc: Planilha de Cálculo**
- **Draw: Editor de Imagens**
- **Math: Manipulação de fórmulas matemáticas**

Fonte: IFRN (2013)

Processos

Um processo é um canal de comunicação entre os programas que estão sendo executados no sistema operacional e o usuário, ou seja um processo é um programa que está sendo executado.

Processos

No Linux / Unix os processos utilizam os recursos de forma inteligente diminuindo o uso do processador (CPU), ou seja quando processo de um programa tentar ler ou escrever informações no HD, este processo entrará na fila de espera até finalizar a entrada e saída dados, enquanto isso ocorre, outro processo pode ser executado diminuindo o tempo de uso e acesso do processador, memória RAM e periféricos

Componente de um Processo

A execução de um processo possuem vários componentes, em particular o PID (*Process Identification*) é um número de identificação que o sistema exibe a cada processo. Nenhum PID pode existir para dois ou mais processos ao mesmo tempo, devendo ser gerado um novo numero a cada um novo processo.

Componente de um Processo

Lembrando que cada processo necessita de um proprietário, ou seja, um usuário que seja considerado seu dono. Nesse ponto, o sistema irá identificar, quem pode ou não executar o processo em questão, isso de acordo com as permissões do usuário. Para lidar com os donos, o sistema utiliza os números UID (*User Identifier*) e GID (*Group Identifier*).

Gerenciamento de Tarefas (Jobs)

É a maneira pelo o qual os processos são finalizados e retornados ao seu estado inicial, os processos são executados no terminal de comandos através de comandos de *shell* pelo usuário.

No exemplo abaixo é executado o programa **firefox em** backgroung **&.**

$ firefox &

[1] 3424

$ jobs –l

[1] + 3424 concluído firefox

Gerenciamento de Tarefas (Jobs)

É a maneira pelo o qual os processos são finalizados e retornados ao seu estado inicial, os processos são executados no terminal de comandos através de comandos de *shell* pelo usuário.

No exemplo abaixo é executado o programa **firefox em** backgroung **&.**

$ firefox &

[1] 3424

$ jobs –l

[1] + 3424 concluído firefox

Gerenciamento de Tarefas (Jobs)

Acima foi exibido o número do *job* **[1] e o PID do processo 3424.**

E se fosse para o processo **xeyes?**

Comando Job e suas opções

jobs - Exibe os jobs em execução

Opções:

-l **Exibe o nome e o número de cada processo**
-s **Exibe o nome de cada processo**
-p **Exibe o número de cada processo**

Exemplo:

$ xcalc &
$ jobs
$ jobs -l
$ jobs -p

Comando **ps** e suas opções

ps - Mostra informações sobre processos em execução

Opção:

-A	Mostra todos os processos
-a	Mostra informações de outros usuários
-u	Mostra o nome do usuário e o horário de início do processo
-x	Mostra processos do terminal corrente e de outros terminais
-p	Mostra o número do processo PID

Exemplo:

$ firefox &

$ ps -aux

Resultado exibidos

USER	Nome do usuário dono do processo
PID	ID do processo
%CPU	Porcentagem do consumo de CPU
%MEM	Porcentagem do consumo de memória
VSZ	Tamanho virtual do processo
RSS	Número de páginas na memória (*Resident Set Size*)
TTY	TTY do terminal
STAT	Estado processo: **R (executável), S (Dormente (< 20 segundos)), Z** (Zumbi), **D (espera no disco(curto prazo)), T (rastreado** ou interrompido)

Resultado exibidos

Outras opções adicionais:

START	Horário de início do processo
TIME	Tempo de consumo do processador (CPU) consumido pelo processo
COMMAND	Nome do comando e opções

Comando **pstree**

pstree - Mostra informações sobre processos em execução em forma de árvore

Opções:

-a Mostra opções de comandos no terminal .
-c Não compacta subárvores.
-l Mostra linhas de forma detalhada
-n Classifica os processos pelo PID
-p Exibe o PID dos processos.

Exemplo:

$ pstree -p
$ pstree -nap

Comando **top**

top - Visualiza os processos com o maior uso do processador (CPU)

Opções:

-u Mostra processos de um usuário específico
-U Não mostra processos de um usuário específico
-d (n) Atualiza a tela do monitor a cada n segundos

Exemplo:

$ top –u aluno
$ top –u aluno –d 4

Comando **kill**

kill - Finaliza um processo pelo PID

Para matar um processo são utilizadas 3 maneiras usando os sinais negativos:
OBS: É necessário estar como root.

-9, -SIGKILL e -KILL.

Exemplo:
$ ps –A
$ kill -9 4445

Comando **man**

Traz um manual completo sobre os comando linux.

Exemplo:

$ man ls

Exercícios

1) Executar três processos (gimp, gedit e bluefish) em background e para cada um desses aplicar as opções do comando **jobs** e **ps**. Estruturar no *LibreOffice Writer* uma tabela com os resultados obtidos. (Caso necessário instalar um dos programas mencionados utilizar o comando sudo apt-get install NomePrograma)
2) Apresentar todos os processos em execução em forma e árvore, classificando-os pelo PID. Apresentar os resultados no *LibreOffice Writer .*
3) Mostrar no terminal *linux* os processos que estão consumindo mais processador, atualizar a listagem a cada 5 segundos.

Exercícios

4) Criar um diretório em /home/aluno/Documentos com pelo menos cinco arquivos. Em seguida, ordenar esses arquivos pela hora da última modificação dos mesmos e posteriormente ordenar por tamanho do arquivo. Para isso utilize o comando **man** para **ls**, descobrindo quais as opção do comando **ls** que permitem ordenar por última modificação e tamanho de aquivo.
5) Colar em execução pelo menos dois processos usuais (editor de texto, calculadora e etc). Listar no terminal do linux todos os processos em execução. Finalizar dois processos e novamente mostrar a lista de processos em execução. Documentar essas etapas no LibreOffice Writer.