

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROGRAMAÇÃO DE COMPUTADORES I

Prática 01: Conceitos Iniciais / Sistema Operacional

Sistema Operacional Linux – Comandos Básicos



- A atividade prática aqui proposta deve ser efetuada através da utilização de um terminal Linux.

Sistema Operacional Linux

Os sistemas UNIX/Linux são caracterizados por serem:

- **Interativo** – usuário requisita os comandos e obtém os resultados de sua execução através do terminal;
- **Multitarefa** – um mesmo usuário pode efetuar vários comandos em paralelo no seu terminal, ficando a cargo do sistema operacional efetuar o controle destas execuções simultâneas.
- **Multiusuário** – vários usuários podem utilizar terminais diferentes, sendo tarefa do sistema operacional controlar as requisições de comandos de cada um deles e distribuir de forma correta os recursos de hardware necessários a cada usuário.

Terminal: Comandos Básicos Linux

Um comando é um arquivo (programa executável) armazenado em um determinado diretório do sistema. Ao ser executado, funciona como qualquer outro programa do sistema quando acionado.

A sintaxe mais comum é **comando [opções] [argumentos]**, onde:

- **comando** – é o nome do comando ou programa a ser executado;
- **opções** – são modificadores do comando;
- **argumentos** – complemento ao comando ou suas opções.

Exemplo: **top -d 5** (**“top”**: nome do comando)
 (**“-d”**: opções do comando)
 (**“5”**: argumentos do comando)

Obs.: o comando **“top”** permite visualizar o estado dos processos em execução dentro de um sistema Linux, a opção **“-d”** indica que queremos atualizar o estado dentro de um determinado tempo em segundos, e o argumento **“5”** especifica qual o valor desse tempo.

1. Navegação pelos Diretórios

- **"pwd"** – indica qual o diretório atual do *shell*.
- **"cd dir"** – muda para o diretório dir.
- **"cd .."** – muda para o diretório pai imediatamente superior.
- **"cd -"** – volta para o último diretório visitado.
- **"cd ~"** – volta ao diretório HOME.
- **"mkdir dir"** – criação do diretório dir.
- **"rmdir dir"** – remoção do diretório dir.

2. Processos do Sistema

- **"ps -x"** – possibilita visualizar as informações de todos os processos em execução no sistema.

3. Operações básicas com arquivos

- **"ls"** – listar o conteúdo do diretório corrente (ou de um diretório dado).
- **"rm"** – serve para remover um arquivo do sistema. Ele só fará a remoção de diretórios se for utilizada a opção **-r**.
- **"mv"** – permite movimentar um arquivo ou um diretório de um local para outro dentro do sistema.
- **"cp"** – permite copiar arquivos e diretórios. O comando funciona de três formas básicas: 1) copia o conteúdo de arquivos para dentro de outros arquivos; 2) copia arquivos para dentro de diretórios; 3) copia o conteúdo de diretórios para dentro de outros diretórios.
- **"cat"** – apresentar o conteúdo de arquivos.

Repositório Linux: (apt-get install)

Considerando que estamos utilizando versões Linux derivadas do Debian, como por exemplo, Ubuntu:

- **sudo su**
- **apt-get install kolourpaint**
- **apt-get install gimp**

Gerenciamento de Tarefas



- A atividade prática aqui proposta deve ser efetuada através da utilização de um terminal Linux.

Comando Kill

O comando **"kill"** permite modificarmos o estado interno de uma tarefa, como visto na parte teórica da aula. Sendo assim, através do uso do comando **"kill"** é

possível transitarmos uma tarefa entre os estados: **executando** / **suspensa** / **terminada**.

- **kill -STOP [PID]** – permite suspender a execução da tarefa com número de identificação igual a PID. (vai para o estado: *suspensa*).
- **kill -CONT [PID]** – permite continuar a execução da tarefa com número de identificação igual a PID, que estava no estado de *suspensa*. (vai para o estado: *executando*).
- **kill -TERM [PID]** – permite finalizar a execução do processo com número de identificação igual a PID. (vai para o estado: *terminada*).

Cálculo de π com Threads



- A atividade prática aqui proposta deve ser efetuada através da utilização de um terminal Linux, pela compilação e execução do código-fonte disponível no link: <http://www.gileduardo.com.br/ifpr/so/downloads/calcp.c>

Compilação e Execução do código-fonte C (calcp.c)

Primeiramente, após efetuar o download do arquivo, descompacte-o, identifique o local onde está salvo na máquina, abra um terminal Linux, e utilize o comando “**cd**” para navegar até o diretório onde se encontra o arquivo **calcp.c**.

Após isso faça a compilação e execução do código como mostrado a seguir:

- **Compilação:** `gcc calcp.c -o pi`
- **Execuções:** `time ./pi 1`
`time ./pi 2`
`time ./pi 4`

Comunicação entre Tarefas



- A atividade prática aqui proposta deve ser efetuada através da utilização de um terminal Linux.

Pacote “netcat”

O pacote **netcat** permite que duas tarefas se comuniquem através da utilização do conceito de *sockets*. Sendo assim torna-se possível montar um canal de comunicação entre duas tarefas, em uma mesma máquina, ou em máquinas distintas.

Comunicação entre tarefas locais

- **Instalação do pacote:** `apt-get install nc` *(distribuições Debian)*
- **Execução do servidor:** `nc -l 12000`
- **Execução do cliente:** `nc localhost 12000`

Comunicação entre tarefas remotas

- **IP servidor:** `ipconfig`
- **Execução do servidor:** `nc -l 12000`
- **Execução do cliente:** `nc ip_servidor 12000`

** (ip_servidor deve ser substituído pelo IP da máquina servidor)*