

## Lista Prática 1 – Ambiente Linux

- 1) Antes de começarmos, vamos preparar o ambiente para a prática. Siga os passos abaixo:
  - Abra o terminal do Ubuntu (CTRL + ALT +T)
  - Digite o comando “ls” (sem aspas), sua função é listar o conteúdo de um diretório.
  - Crie uma pasta para você salvar os arquivos da prática. Para isso utilize o comando “mkdir Lab01”, este comando vai criar uma pasta chamada Lab01.
  - Digite “ls” e veja se você consegue visualizar a nova pasta criada.
  - Após criar a pasta, digite “cd Lab01” para acessar o conteúdo da pasta.
  - Acesse o moodle e baixe o arquivo “codes.zip”. Salve-o na pasta “cd Lab01”
  - Digite o comando “unzip codes.zip” para descompactar os arquivos.
  - Digite o comando “cd codes” para acessar os arquivos
  - Caso tenha alguma dúvida sobre os parâmetros de um comando, digite “man” e depois o nome do comando para ver a documentação.
- 2) Acesse o diretório onde você salvou os arquivos de “codes.zip”. O arquivo p1.c apresenta um programa básico em C que contém um loop infinito.
  - Procedimento 1:
    1. Abra o terminal do Ubuntu (CTRL + ALT +T)
    2. Execute o código-fonte
    3. Para compilar digite: gcc p1.c -o p1
    4. Execute o comando: ./p1
    5. Tente digitar o comando ps -lA
    6. Abra um novo terminal (CTRL + ALT + T)
    7. Digite o comando ps -lA
    8. Anote o PID do processo referente a “p1”
    9. Digite o comando kill PID (substitua pelo valor no obtido no passo anterior)
    10. Digite novamente o comando ps -lA

Responda as questões a seguir:

- a) O que aconteceu após executar o programa no passo 4? A tela do terminal ficou travada na execução do programa? Você conseguiu digitar outros comandos?
  - b) O que faz o comando ps -lA?
  - c) O que aconteceu com o processo relacionado a p1, após a execução do procedimento 1? Qual foi a ação ocasionada pelo comando kill?
- Procedimento 2:
    - 1) Abra o terminal do Ubuntu (CTRL + ALT +T)
    - 2) Execute o código-fonte
    - 3) Execute o comando: ./p1&
    - 4) Digite o comando ps -lA no mesmo terminal

Responda as questões a seguir:

- d) O que aconteceu após executar o programa? A tela do terminal ficou travada ou foi possível executar outros comandos normalmente?
- e) O que faz o comando &?
- f) Porque na realização do procedimento 1 o comportamento da tela do terminal associado a execução de p1 teve um comportamento diferente?

3) Pipes: Acesse o diretório onde você salvou os arquivos de “codes.zip”.

- Procedimento 1:
  1. Abra o terminal do Ubuntu (CTRL + ALT +T)
  2. Acesse o diretório em que você salvou os arquivos
  3. Digite o comando cat c1 e analise os dados
  4. Digite o comando cat c2 e analise os dados
  5. Digite o comando cat c1 c2 | sort > out.txt
- Procedimento 2:
  1. Abra o terminal do Ubuntu (CTRL + ALT +T)
  2. Acesse o diretório em que você salvou os arquivos
  3. Digite o comando cat c1 e analise os dados
  4. Digite o comando cat c2 e analise os dados
  5. Digite o comando cat c1 c2 | grep Ruiz
  6. Digite o comando cat c1 c2 | grep Florinda

Responda as questões a seguir:

- a) O que aconteceu após realizar este procedimento 1 e 2?
- b) Explique em termos teóricos (teoria de processos) o que ocorreu na etapa 5 do procedimento 1 e nas etapas 5 e 6 do procedimento 2.

4) Exemplo de Subprocesso. Acesse o diretório onde você salvou os arquivos de “codes.zip”.

- Procedimento 1:
  1. Abra o terminal do Ubuntu (CTRL + ALT +T)
  2. Digite nano p2.c para verificar o código-fonte. Utilize o comando “CTRL + X” para sair do editor.
  3. Compile o código-fonte: gcc p2.c -o p2
  4. Execute o código: ./p2

Responda as questões a seguir:

- a) Ao analisar o código-fonte, percebe-se que foi declarado e inicializado um vetor de inteiros, de forma que, o processo filho deveria atribuir os valores 10 e 20 para as posições 2 e 3 e o processo pai, 30 e 40 para as posições 0 e 1. No entanto, ao final da execução do processo pai e de seu filho, o conteúdo do vetor não foi: [30,40,10,20], porquê?