1. Como se utiliza o comando ps para:

(a) Mostrar todos os processos rodando na máquina?

$ ps -ef

(b) Mostrar os processos de um usuário?

$ ps -f -u dariodes

(c) Ordenar todos os processos de acordo com o uso da CPU?

$ ps aux --sort=-pcpu

(d) Mostrar a quanto tempo cada processo está rodando?

$ ps -e -o pid,comm,etime

1. De onde vem o nome fork()?

Significa bifurcar.

1. Quais são as vantagens e desvantagens em utilizar:

(a) system()?

Permite executar um comendo dentro de um programa, criando um sub-processo (processo filho).

O uso do system() não é recomendado na maioria dos casos. É simples mas dá brechas e falhas de execução, pois depende muito do sistema.

(b) fork() e exec()?

No Unix não há meio para que um processo crie outro somente mandando executá-lo diretamente. Não há função qe crie e execute u novo processo em um único passo. Para isso, utiliza-se a função fork() que cria uma cópia do processo atual e em seguida a função exec() que substitui o conteúdo do novo processo por um novo programa.

1. É possível utilizar o exec() sem executar o fork() antes?

Sim, mas o resultado não é o desejado.

1. Quais são as características básicas das seguintes funções:

(a) execp()?

Aceita que o nome ou procurado programa esteja no current path.

(b) execv()?

Aceita que alista de argumentos do novo programa seja nula.

(c) exece()?

Aceita um argumento adicional.

(d) execvp()?

Aceita que alista de argumentos do novo programa seja nula e aceita que o nome ou procurado programa esteja no current path.

(e) execve()?

Aceita que alista de argumentos do novo programa seja nula e aceita um argumento adicional.

(f) execle()?

Aceita em sua lista de argumentos a utilização de mecanismos var args em Linguagem C e aceita um argumento adicional.