

LISTA 2

1) Explique a função dos Sistemas de Arquivos. Adicionalmente, descreva a diferença entre o sistema de arquivos do linux e do windows.

Para que uma partição de disco possa ser usada para armazenar dados, primeiro ela deve ser formatada. O processo de formatação inclui “carimbá-lo” com um sistema de arquivos.

Um sistema de arquivos é um conjunto de estruturas lógicas e de rotinas, que permitem ao sistema operacional controlar o acesso ao disco rígido. Conforme cresce a capacidade dos discos e aumenta o volume de arquivos e acessos, esta tarefa torna-se mais e mais complicada, exigindo o uso de sistemas de arquivos cada vez mais complexos e robustos.

Dependendo da escolha e o uso do sistema, pode haver lentidões e outros problemas. Assim, dentre diversos sistemas de arquivos Linux existentes, existe um que se encaixa melhor com seu perfil de uso, e proporciona melhores condições de trabalho. Existem diversos sistemas de arquivos diferentes, que vão desde sistemas simples como o FAT16, até sistemas como o EXT4, BtrFS e ZFS, que incorporam recursos muito mais avançados.

O sistema de arquivos no Windows é NTFS (New Technology File System). Já o Linux, é conhecido por ter uma gama de sistemas de arquivos suportados, inclusive que são suportados em outros sistemas operacionais. Existem muito desde dos mais simples, os experimentais e os que podem ser o futuro da próxima geração de sistemas de arquivos para Linux.

2) Existem quatro tipos de problemas que podem ocorrer na execução de processos concorrentes: trancamento (lockout), impasse (deadlock), inanição (starvation) e indeterminismo. Explique cada um deles dando exemplos de situações onde podem ocorrer.

Lockout: Quando vários processos estão aguardando para acessar um recurso compartilhado por exemplo.

Deadlock: Ocorre quando os processos ficam parados sem possibilidade de iniciar ou continuar seus processamentos. Ocorre normalmente quando um conjunto de

processos está esperando por um evento que somente um outro processo desse mesmo conjunto poderá fazer acontecer.

Starvation: Quando um processo nunca consegue acessar o recurso compartilhado. Ocorre se o processo tiver baixa prioridade ou o acesso ao recurso for aleatório, nestes casos existem grandes chances dele nunca ser executado.

Indeterminismo: múltiplas execuções de uma tarefa podem não gerar o mesmo resultado.

3) Faça um programa que imprima os números primos existentes entre 0 e 99999. UTILIZE THREADS. Dica: para cada faixa de mil valores crie uma thread e dispare o processo para cada uma delas.

4) Implemente um programa que simule um lista de tarefas, usando listas encadeada por meio da biblioteca linux/list.h. Ver <https://github.com/torvalds/linux/blob/master/include/linux/list.h>