

Lista 02 de Sistemas Operacionais

Professor: Herbert Rocha
Aluno: William Juan da Silva Melo

Respostas

1) A função do sistema de arquivos é definir como os arquivos serão organizados na memória. O sistema dita as regras de como eles são armazenados, organizados, buscados e movidos dentro de toda a dinâmica de dados que ali acontece.

A diferença do sistema de arquivos do Linux e do Windows começa já pela liberdade. No Windows estamos presos ao sistema NTFS, sistema esse criado a muito tempo atrás pela Microsoft e ainda mantido por causa da compatibilidade entre as versões mais novas do Windows e as versões mais antigas. Há o FAT32 também, geralmente reservado à mídias removíveis, e também entendido pelo Windows. Já no Linux, voltando o que falei sobre liberdade, temos a possibilidade de escolher entre diversos sistemas de arquivos open-source, como a família ext, o xfs, o reiser, entre outros, cada um com suas qualidades e defeitos. O mais usado hoje em dia é o ext versão 4, ou simplesmente ext4, uma evolução do ext3. Uma de suas principais características, e que lhe dá bastante vantagem em relação ao NTFS do Windows, é sua forma organizada de armazenar os dados, que quase não desfragmenta o disco, por isso o geralmente os sistemas Linux não tem a necessidade de usarem um desfragmentador para manter a boa velocidade do sistema.

2)

Trancamento (lockout): é quando a tarefa fica aguardando um evento que nunca ocorrerá. Um exemplo é um cliente que aguarda dados de um servidor que não pode lhe enviar estes dados. Este cliente não pode ficar aguardando para sempre.

Impasse (deadlock): é quando os processos necessitam de um mesmo recurso ao mesmo tempo, o que faz ocorrer um “engarrafamento” na escolha de qual processo usará o recurso primeiro. Um exemplo são vários processos ou várias threads usando um dado que está em um espaço de memória, e todos eles estão necessitando desse dado ao mesmo instante, qual será atendido primeiro? Esse problema pode ser corrigido por recursos de exclusão mútua.

Inanição (starvation): É quando um processo com menor prioridade necessita de um recurso, mas outros processos de maior prioridade usam esse recurso por muito tempo, faz o processo menor nunca ter a oportunidade de usá-lo também, por isso dizemos que ele “morre de fome”. Uma forma simples de resolver esse problema é criando uma fila de prioridades, onde processos de maior prioridade terão a preferência pelo recurso, mas os processos menores também terão uma chance.

Indeterminismo: múltiplas execuções de uma mesma tarefa podem não gerar o mesmo resultado. Isso pode ocorrer, por exemplo, se o processo usa dados que podem ser alterados externamente, por outro processo.