

Universidade Federal de Roraima
Departamento de Ciência da Computação

DISCIPLINA: Sistemas Operacionais – DCC403

Tiago Araújo Nunes de Lima

Lista II

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

O **sistema de arquivos** é a parte do sistema operacional que lida com arquivos. Como são estruturados, nomeados, acessados, usados, protegidos, implementados e gerenciados. Existem muitas diferenças entre o sistema de arquivos do Windows e do Linux.

Vejamos no Windows primeiro, o sistema de arquivos não distingue letras maiúsculas e minúsculas, ou seja linux, Linux e LINUX são nomes que se referem ao mesmo arquivo. No Linux seriam arquivos diferentes. Os tipos de arquivos também diferem, cada um tem seu próprio executável. No Windows a extensão dos arquivos deve ter entre 1-3 caractere, já no Linux é facultado podendo ter até mais de uma extensão. O diretório raiz no Linux é o / e os componentes do caminho são separados por /, no Windows seria C:\ (ou qualquer letra) e separando os diretórios com \.

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.

Indeterminismo: quando dois processos concorrentes acessam um recurso compartilhado ao mesmo tempo. Sendo global a variável data, estas funções executadas concorrentemente podem gerar indeterminismo durante o escalonamento.

int data = 0;

```
void *inc(){  
    int a = data;  
    a = a + 1;  
    data = a;
```

```
}
```

```
void *dec(){  
    int b = data;  
    b = b - 1;  
    data = b;
```

```
}
```

Trancamento (lockout): vários processos um evento, ou seja, um recurso ocupado por outro processo, diminuindo o desempenho. Um exemplo seria o bloqueio ao acesso de outros processos a variável data do exemplo anterior, executando um e depois o outro.

Impasse (deadlock): quando dois (ou mais) processos esperam um recurso que só pode ser produzido pelo outro. Por exemplo, um processo A, utilizando um recurso P, necessita

de um recurso Q, alocado pelo processo B, para continuar o processamento e o processo B necessita do recurso P para continuar o processamento.

Inanição (starvation): quando um processo com baixa prioridade nunca recebe acesso ao recurso, pois outros processos de prioridade mais elevada passam na sua frente. Um exemplo poderia ser um servidor de impressão que tenha a regra de prioridade por menor quantidade de páginas. Assim a medida que novas impressões menores são inseridas na fila, impressões maiores podem nunca ser impressas.

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.

https://github.com/tiago-araujo/Tiago_Lista_2_os_rr_2018.git