

**Universidade Federal de Roraima**  
**Departamento de Ciência da Computação**

DISCIPLINA: Sistemas Operacionais – DCC403

**LISTA 2 - Prazo de Entrega: 13/06/2019**

ALUNO(A): **FRANCISCO PEREIRA DO NASCIMENTO**

NOTA: \_\_\_\_\_

**ATENÇÃO:** Descrever as soluções com o máximo de detalhes possível, no caso de programas (escritos em C ou C++), inclusive a forma como os testes foram feitos. Todos os artefatos (relatório, código fonte de programas, e outros) gerados para este trabalho devem ser adicionados em um repositório no site [github.com](https://github.com), com o seguinte formato `nomeDoAluno_Lista_2_os_rr_2019`.

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

É a forma de organização de dados em algum meio de armazenamento de dados em massa, frequentemente feito em discos magnéticos. Ele controla como os dados são armazenados e recuperados, possibilitando ao sistema operacional decodificar os dados armazenados e lê-los ou gravá-los. Sem um sistema de arquivos, as informações colocadas em um meio de armazenamento seriam um grande corpo de dados, sem nenhuma maneira de dizer onde uma parte da informação termina e a próxima começa. Ao separar os dados em pedaços e dar um nome a cada peça, a informação é facilmente isolada e identificada. Tomando seu nome a partir do modo como os sistemas de informação baseados em papel são nomeados, cada grupo de dados é chamado de "arquivo". A estrutura e as regras lógicas usadas para gerenciar os grupos de informações e seus nomes são chamadas de "sistema de arquivos".

**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**

**O que é:**

Vários processos aguardando um evento (recurso ocupado), reduzindo o desempenho. Quando algum processo tenta acessar a seção crítica já ocupada, esse fica em lockout, travado, até que o espaço seja desocupado.

**Universidade Federal de Roraima**  
**Departamento de Ciência da Computação**

**DEADLOCK-**

**O que é:**

**Quando ocorrem:**

**Exclusão mútua** - Um recurso só pode estar alocado para um processo em um determinado momento.

**Uso e espera** - Processos que já possuem algum recurso podem requerer outros.

**Não preempção** - Recursos já alocados não podem ser retirados; só o próprio processo pode liberá-los.

**Espera circular** - Um processo espera por recursos alocados a outro processo, em uma cadeia circular.

**Exemplo:**

Dois processos querem gravar em CD um documento obtido pelo scanner. O processo A está usando o scanner, enquanto o processo B, que é programado diferentemente, está usando o gravador de CD. Então, o processo A pede para usar o gravador de CD, mas a solicitação é negada até que o processo B o libere. Porém, ao invés de liberar o gravador de CD, o processo B pede para usar o scanner. Nesse momento, ambos os processos ficam bloqueados e assim ficarão para sempre.

**STARVATION**

**O que é:**

Em programação concorrente, ocorre inanição quando um processo nunca é executado ("morre de fome"), pois processos de prioridade maior sempre o impedem de ser executado

A ocorrência da inanição se dá quando os programas rodam indefinidamente (razão pela qual também se dá o nome de preterição indefinida a esta situação) e não fazem nenhum progresso em seu processamento, ao contrário do deadlock, que ocorre quando os processos permanecem bloqueados, dependendo da liberação dos recursos por eles alocados.

**Quando ocorrem:**

Quando dois ou mais processos esperam o mesmo recurso, mas um não nunca conseguir usar tal recurso.

**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.**

**Universidade Federal de Roraima**  
**Departamento de Ciência da Computação**

**4) Implemente um programa que simule um lista de tarefas, usando listas encadeada por meio da biblioteca linux/list.h.**

```
> make  
> sudo insmod llist.ko  
> dmesg
```