



# Universidade de Brasília (UnB)

## Departamento de Ciência da Computação

**Disciplina:** Sistemas Operacionais

**Prof:** Alba Melo

**Período:** 02/2022

### Descrição do Trabalho Prático (02/2022)

#### 1. Pré-requisitos

O aluno deverá conhecer a linguagem de programação C e chamadas de sistema Unix.

#### 2. Descrição

O projeto consiste da confecção de um escalonador de processos CPU-bound em modo usuário. Cada processo CPU-bound deve usar cerca de 30 segundos de tempo de CPU.

Existirá um processo rodando em *background* (*execprocessod*), que implementará política de escalonamento com prioridades dinâmicas com 3 filas (alta, média, baixa) onde cada fila é regida pela política Round-Robin (RR) e possui o seu quantum (10 segundos).

##### 2.1 Submissão de Processos

O processo *execproc* receberá o arquivo a ser executado e seus parâmetros (se existirem), bem como a prioridade inicial e os colocará na fila *ready*. Será repassado ao processo que solicitou a execução um identificador do processo.

Sintaxe:

```
> execproc <arquivo-executável> <prioridade> [<param1> <param2> ... <paramn>]
```

##### 2.2 Execução de Processos

A execução dos processos será feita por um processo em *background* (*execprocd*), que implementará a política de escalonamento. Seu funcionamento é o seguinte. Para cada processo P no início da fila de maior prioridade não vazia, o processo *execprocd* colocará em P execução pelo tempo estipulado no seu quantum. Quando o quantum se esgotar, o processo terá sua prioridade recalculada e irá para o final da respectiva fila e o processo no início da fila de maior prioridade será colocado em execução. Quando o processo P terminar sua execução, é impresso o tempo total de execução deste processo (wallclock time – tempo de turnaround), o número de trocas de contexto e seu identificador.

##### 2.3 Cancelamento da Execução

A qualquer momento, o processo pode ter a sua execução cancelada através do comando *cancela\_proc <id\_proc>*. Ao ser digitado esse comando, o processo, caso esteja executando, perde a CPU e os dados de final de execução são impressos. Caso o processo esteja na fila *ready*, ele é retirado da mesma e os dados de final de execução são impressos.

### **2.4 Shutdown do processo *execprocd***

O processo *execprocd* terá sua execução terminada através do comando *shutdown\_execprocd*. Nesse momento, todos os processos em execução são terminados e é impresso em tela um relatório contendo o número total de processos executados, o número de processos cancelados e o número total de trocas de contexto ocorridas.

### **3. Processos envolvidos**

Criar pelo menos 4 programas:

- a) *execproc*: solicita a execução dos programas e imprime o identificador na tela
- b) *execprocd*: processo em *background* que controla a execução de processos
- c) *cancela\_proc*: solicita o cancelamento da execução dos processos
- d) *termina\_execprocessod*: termina a execução dos processos

Os programas a serem executados no nosso sistema devem ser *CPU/bound* e ter duração de execução maior que 30s.

### **4. Algoritmos de determinação da prioridade**

Os algoritmos de determinação da prioridade serão os seguintes: (a) escolha randômica; (b) prioridade estática; (c) prioridade dinâmica (os alunos devem propor e implementar o algoritmo).

### **5. Sugestões**

A execução de um processo pode ser interrompida e depois retomada através dos sinais SIGTSTP e SIGCONT, respectivamente.