



Bacharelado em Ciência da Computação
Trabalho 2 – Algoritmos e Estrutura de Dados II (Prática)
Prof. Fellipe Rey

Objetivo:

Utilizar a estrutura de dados pilha, fila e lista (POR ALOCAÇÃO DINÂMICA) na resolução de um problema de programação.

Procedimento:

Leia com atenção o enunciado dos problemas abaixo, desenvolva um algoritmo e implemente o programa em C para resolver o problema descrito. Após a implementação, teste o programa várias vezes para os diferentes dados que foram gerados.

Problema:

Com o intuito de executar vários processos tendo à disposição uma quantidade limitada de hardware (processador), o computador possui um **escalonador**, que é responsável por definir qual processo será executado em um determinado intervalo de tempo.

Existem vários métodos de escalonamento diferentes. Alguns deles são o **First Come, First Served** (FCFS, em português: Primeiro a chegar, primeiro a ser servido), **Shortest Job First** (SJF, em português: Trabalho mais curto primeiro) e o **Round-Robin** (escalonamento por alternância circular).

Roteiro:

1. Desenvolva um programa que implementa os métodos FCFS, SJF e Round-Robin utilizando as estruturas de dados aprendidas em sala de aula – fila, lista e lista, respectivamente.
2. No início do programa, o usuário deverá escolher qual método de escalonamento será executado.
3. A cada iteração do programa, ele poderá (chance de 30%) criar um processo, que terá o tamanho entre 1 e 20 (somente números inteiros). Este processo deverá ser armazenado na Estrutura de Dados correspondente ao método de escalonamento escolhido.
4. Cada iteração do programa corresponderá a uma unidade de tempo do sistema. Se um processo foi criado com tamanho 5, isso significa que a aplicação deverá gastar 5 iterações (ou seja, cinco “voltas” no laço de repetição) para concluir o seu processamento.
5. No caso dos métodos de escalonamento FCFS e SJF, a execução do próximo processo só ocorrerá ao término da execução do processo atual. No caso do Round Robin, ele executará com um *quantum* de 6 unidades de tempo. Caso o processo termine em 6 unidades de tempo (ou menos), ele é removido da lista e o próximo elemento é executado. Caso o processo tenha um tempo de execução superior à 6 unidade de tempo, ele volta ao término da lista, descontando o tempo processado (exemplo: Um processo de tamanho 10 é executado por 6 unidades de tempo e volta ao fim da lista com 4 unidades de tempo restantes).
6. Instruções do funcionamento de cada um dos métodos estão no arquivo “(Exemplo) Escalonamento de Processos” fornecido no Moodle.

Observação:

1. Inclua um comentário no cabeçalho de cada programa fonte com o seguinte formato:

```
(*+-----+
|          UNIFAL - Universidade Federal de Alfenas.          |
| BACHARELADO EM CIENCIA DA COMPUTACAO.                      |
| Trabalho...: Validacao de arquivos XML                     |
| Disciplina: Algoritmos e Estrutura de Dados II - Pratica    |
| Professor.: Fellipe Rey                                     |
| Aluno(s)...: Fulano da Silva                                |
|              Beltrano da Silva. (MAXIMO 4 ALUNOS).          |
| Data.....: 99/99/9999                                       |
+-----+*)
```

2. Enviar num arquivo único (.ZIP) através do Envio de Arquivo do MOODLE.