T1-SISOP

Grupo: Adilson Medronha e Luís Lima

Instruções para execução

- O arquivo de entrada para o programa é o `input.json` e nele é possível colocar as configurações, como:
  - **quantum:** quantum do sistema
  - **memory:** capacidade das filas de **blocked** e **ready**
  - **processes:** lista contendo as informações do processo como:
  - **filename:** *path* do arquivo que contém o código do processo
  - **priority:** prioridade do processo
    - **DEVE SER NECESSÁRIAMENTE UMA DAS SEGUINTES (EM CAPSLOCK): HIGH, MEDIUM ou LOW**
  - **arrival_time:** tempo de chegada do processo (necessáriamente inteiro)
  - O código já está configurado para receber o arquivo, portanto não é necessário fazer nenhuma alteração nele, apenas no arquivo de configuração.
- Após configurado o arquivo de entrada, basta executar o comando `python3 os.py` e o programa irá iniciar mostrando as filas.

Instruções durante a execução

- O programa inicia mostrando as filas com os processos passados por arquivo de configuração.
- **A aparição do símbolo `>>` no terminal indica que o usuário deve apertar `ENTER` para seguir com a execução do programa**
  - A cada iteração o programa irá imprimir as filas, qual instrução está sendo executada, o valor da variável acumulador e o tempo global da execução.
- **Ao final da execução, o programa irá imprimir uma tabela contendo as seguintes informações de cada processo:**
  - *ARRIVAL TIME*
  - *START TIME*
  - *END TIME*
  - *PROCESS TIME*
  - *WAITING TIME*
  - *TURNAROUND TIME*

Informações de implementação

- O trabalho, em geral, não trata todas as questões de *fairness* pois prioriza **sempre** processos com maior prioriodade, podendo ocorrer inclusive um monopólio de processador.
- O arquivo `process.py` contém a modelagem de um **processo**. Em seus atributos, há informações importantes para a execução do programa, como:
  - **Informações do Processo**
  - PID
  - Instruções
  - Área de dados
  - Prioridade
  - Área de labels
  - Estado
  - **Informações utilizadas para controle do bloco de processos (PCB) (salvamento e retomada de contexto)**
  - Último valor do acumulador
  - Último valor de PC
  - *Último PC que aponta para uma label*
    - Detalhe de implementação para tratar os saltos condicionais
  - **Informações utilizadas para estatísticas**
  - *Arrival time*
  - *Process time*
  - *Blocked time*
    - Aleatório entre 10 a 20 *u.t*
  - *Start time*
  - *End time* (foi para *EXIT*)
  - *Turnaround time*
- No arquivo `os.py` é onde de fato se encontra o escalonador de processos
  - Leitura do arquivo de entrada (configuração)
  - *Parsing* da área de código e dados
  - **Escalonamento das filas (*READY* e *BLOCKED*)**
  - Feito no método `do_state_change`
    - O *round-robin* ocorre no método `handle_queues`, na linha **412**, onde é feito um append (adiciona no final da fila) na fila (de blocked ou ready).
    - Entretando, logo na linha abaixo é feito um ordenamento da fila onde o de maior prioridade fica na cabeça da fila (aqui ocorre a injustiça).
  - **Execução dos processos**
  - **Execução das intruções dos processos**