

Relatório de Implementação do Algoritmo Round Robin com Feedback

Arthur Valls - 120177470
Artur Amaral - 122032113
Tales Moreira - 119047549

Grupo 15

Novembro de 2023

1 Introdução

O presente relatório descreve o trabalho elaborado pelo nosso grupo como projeto prático 02 da disciplina, cujo objetivo é construir um programa em linguagem C que simula a execução de um escalonador de processos, assim como feito em um sistema operacional.

Um escalonador de processos é a parte de um sistema operacional que gerencia a ordem de execução dos processos que estão competindo pelos recursos do sistema, especialmente o tempo de CPU. Para tal, ele utiliza algoritmos específicos para determinar a sequência em que os processos serão executados, com base em critérios como prioridade, tempo de execução e outras características.

O programa simula a execução dos processos exibindo na tela o processo que ocupa a CPU a cada unidade de tempo (u.t.), uma medida de tempo arbitrária conhecida pelo sistema computacional.

Conforme novos processos requisitam o uso da CPU, eles são organizados de forma a fornecer uma competição justa, de modo que seja impedida uma competição injusta entre eles, resultando em processo esperando tempo indeterminado por um recurso (starvation), o que geraria um sistema operacional não-responsivo e inconsistente.

2 Comportamento do escalonador

A estratégia de escalonamento implementada é o Round Robin com feedback. O Round Robin é um algoritmo simples e amplamente utilizado, no qual cada processo recebe uma fatia de tempo de CPU fixa antes de passar para o próximo processo. Ao fim da execução, o processo corrente para o final da fila de espera.

No entanto, o Round Robin com feedback introduz uma abordagem mais dinâmica. Esse algoritmo permite que processos que exigem mais tempo de CPU

possam receber fatias maiores do tempo de CPU. A ideia principal é monitorar o comportamento dos processos e ajustar dinamicamente as prioridades.

Cada processo é inicialmente atribuído a uma fila com uma prioridade padrão, no caso do simulador, uma **fila de alta prioridade**. Então os processos na fila são programados usando o algoritmo Round Robin, recebendo fatias de tempo fixas de CPU em uma ordem circular. Se um processo não conseguir ser concluído durante a sua fatia de tempo, ele é movido para uma **fila de baixa prioridade**.

Os processos podem ser interrompidos ao realizarem operações de E/S, pois ficam bloqueados até receberem uma resposta do processo que trata desta operação. No simulador, a depender da E/S realizada, o processo pode voltar quaisquer uma das duas filas iniciais existentes. Existem 3 tipos de E/S no simulador: Disco, fita magnética e impressora.

2.1 Modelo de dados simulador

Segue abaixo a descrição das principais estruturas que são operadas neste simulador. Aliada à explicação do funcionamento do algoritmo, este modelo tornará mais intuitiva o comportamento interno do escalonador, pois ele apenas manipula estes dados de forma a implementar os procedimentos descritos neste relatório.

```
// Definicao do tipo de E/S
typedef enum {
    DISK_IO,
    TAPE_IO,
    PRINTER_IO,
} IOType;

// Definicao dos atributos de uma operacao de E/S
typedef struct {
    IOType io_type;
    int io_duration;
    int start_time;
    int remaining_duration;
} IOOperation;

// Possiveis status que o processo assume neste S.O.
typedef enum {
    READY,
    RUNNING,
    IO
} ProcessStatus;
```

```

// Estrutura que define atributos de um processo.
typedef struct {

    int pid;

    // Dados de tempo
    int arrival_time;      // Instante de chegada
    int burst_time;        // Tempo de CPU
    int end_time;          // Instante de encerramento
    int turnaround_time;   // Tempo de execucao

    // Atributos temporarios.
    int remaining_quantum;
    int remaining_burst_time;
    int current_burst_time;

    // Atributos relacionados as operacoes de E/S deste processo.
    IOOperation* io_operations;
    int num_io_operations;
    int current_io_operation;
    IOType io_type;
    ProcessStatus status;

} Process;

```

2.2 Premissas do simulador

- A macro `MAX_PROCESSES` define o número de processos que serão criados e executados no simulador. No casos da nossa simulação, este valor estava configurado como 5.
- O PID dos processos varia incrementalmente de 0 a `MAX_PROCESSES - 1`.
- A macro `QUANTUM` define a quantidade de unidades de tempo que compõem um quantum. Na simulação, este valor foi 2, pois valores maiores resultam em outputs impraticáveis de escrever no relatório.
- Novos processos podem ter tempo de chegada entre 0 e `MAX_ARRIVAL_TIME`, definida, nesta simulação, para 4.
- O tempo de serviço de um processo varia entre `MIN_BURST_TIME` e `MAX_BURST_TIME`. Na simulação estes valores são, respectivamente, 7 e 10.
- Processos novos entram na fila de alta prioridade.
- Processos que sofreram preempção entram na fila de baixa prioridade.

- Processos que retornam de E/S retornam para a fila ditada pela regra definida para o dispositivo em questão.
- **Descrição das operações de E/S**
 - 1 **Disco** - Duração definida pela macro `DISK_TIME`, configurada para 2 u.t. na simulação. Ao encerrar, o processo que a chamou entra na fila de **baixa** prioridade.
 - 2 **Fita magnética** - Duração definida pela macro `TAPE_TIME`, configurada para 3 u.t. na simulação. Ao encerrar, o processo que a chamou entra na fila de **alta** prioridade.
 - 3 **Impressora** - Duração definida pela macro `PRINTER_TIME`, configurada para 4 u.t. na simulação. Ao encerrar, o processo que a chamou entra na fila de **alta** prioridade.
- Ao início da simulação, um processo pode necessitar realizar de 0 a 2 operações de E/S, sendo a quantidade de operações e tempo de início aleatórios, este definido dentro do intervalo do tempo de execução do programa. O tempo de chegada do processo de E/S é a quantidade de u.t. desde o início da execução de processo, contando apenas o tempo em que o mesmo passou na CPU.

3 Output do simulador


Como o simulador possui um componente aleatório, seu output não é determinístico, portanto, esta saída ilustra apenas uma execução arbitrária do programa.

Para compilar e executar o simulador, anexamos um `Makefile` no diretório fonte onde também se encontram uma pasta `src` e uma pasta `include`, com os códigos em `.c` e os headers em `.h`, respectivamente. Basta utilizar o comando `make` quando dentro do diretório principal onde se encontram os arquivos e pastas mencionados.

Obs: Por questões de formatação, as imagens do output encontram-se anexados nas páginas finais do documento.

Referências

- [1] Brian W Kernighan and Dennis M Ritchie. The c programming language. 2002.
- [2] Carlos A Maziero. Sistemas operacionais: conceitos e mecanismos. *Livro aberto*, 2014.
- [3] William Stallings. *Operating systems internals and design principles*. Prentice-Hall, Inc., 1998.

```
~/repos/Computer-Architecture-and-Operating-Systems/os-scheduler on  master 14:02:38
$ make
gcc -Wall -Wextra -std=c11 -o main src/main.c
./main
===== Processos =====
```

PID	Tempo de serviço	Tempo de chegada	I/O (Tempo chegada)
0	7	3	Disco (2), Disco (2)
1	7	3	Sem I/O
2	8	4	Impressão (3)
3	9	2	Impressão (2), Impressão (3)
4	8	0	Disco (2)

```
===== INSTANTE 0 =====

P4 entrou na fila de alta prioridade.

P4 executou por 1 u.t.

Fila de alta prioridade: P4
Fila de baixa prioridade:
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 1 =====

P4 executou por 1 u.t.
P4 vai para a fila de Disco.

Fila de alta prioridade:
Fila de baixa prioridade:
Fila de I/O Disco: P4
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 2 =====

P3 entrou na fila de alta prioridade.

P3 executou por 1 u.t.
P4 executou 1 u.t. do seu I/O de Disco, faltam 1 u.t.

Fila de alta prioridade: P3
Fila de baixa prioridade:
Fila de I/O Disco: P4
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 3 =====

P0 entrou na fila de alta prioridade.

P1 entrou na fila de alta prioridade.

P3 executou por 1 u.t.
P3 vai para a fila de Impressão.
P4 executou 1 u.t. do seu I/O de Disco, faltam 0 u.t.
P4 finalizou seu I/O de Disco, vai pra fila de baixa prioridade.

Fila de alta prioridade: P0 P1
Fila de baixa prioridade: P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3
```

```

===== INSTANTE 4 =====

P2 entrou na fila de alta prioridade.
P0 executou por 1 u.t.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 3 u.t.

Fila de alta prioridade: P0 P1 P2
Fila de baixa prioridade: P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 5 =====

P0 executou por 1 u.t.
P0 vai para a fila de Disco.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 2 u.t.

Fila de alta prioridade: P1 P2
Fila de baixa prioridade: P4
Fila de I/O Disco: P0
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 6 =====

P1 executou por 1 u.t.
P0 executou 1 u.t. do seu I/O de Disco, faltam 1 u.t.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 1 u.t.

Fila de alta prioridade: P1 P2
Fila de baixa prioridade: P4
Fila de I/O Disco: P0
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 7 =====

P1 executou por 1 u.t.
P1 sofreu preempção, vai pra fila de baixa prioridade.
P0 executou 1 u.t. do seu I/O de Disco, faltam 0 u.t.
P0 finalizou seu I/O de Disco, vai pra fila de baixa prioridade.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 0 u.t.
P3 finalizou seu I/O de Impressão, vai pra fila de alta prioridade.

Fila de alta prioridade: P2 P3
Fila de baixa prioridade: P4 P1 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 8 =====

P2 executou por 1 u.t.

Fila de alta prioridade: P2 P3
Fila de baixa prioridade: P4 P1 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 9 =====

P2 executou por 1 u.t.
P2 sofreu preempção, vai pra fila de baixa prioridade.

Fila de alta prioridade: P3
Fila de baixa prioridade: P4 P1 P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

```

```

===== INSTANTE 10 =====
P3 executou por 1 u.t.
P3 vai para a fila de Impressão.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 11 =====
P4 executou por 1 u.t.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 3 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 12 =====
P4 executou por 1 u.t.
P4 sofreu preempção, vai pro final da fila de baixa prioridade.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 2 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P1 P0 P2 P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 13 =====
P1 executou por 1 u.t.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P1 P0 P2 P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P3

===== INSTANTE 14 =====
P1 executou por 1 u.t.
P1 sofreu preempção, vai pro final da fila de baixa prioridade.
P3 executou 1 u.t. do seu I/O de Impressão, faltam 0 u.t.
P3 finalizou seu I/O de Impressão, vai pra fila de alta prioridade.

Fila de alta prioridade: P3
Fila de baixa prioridade: P0 P2 P4 P1
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 15 =====
P3 executou por 1 u.t.

Fila de alta prioridade: P3
Fila de baixa prioridade: P0 P2 P4 P1
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

```

```

===== INSTANTE 16 =====

P3 executou por 1 u.t.
P3 sofreu preempção, vai pra fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P0 P2 P4 P1 P3
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 17 =====

P0 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P0 P2 P4 P1 P3
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 18 =====

P0 executou por 1 u.t.
P0 sofreu preempção, vai pro final da fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P2 P4 P1 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 19 =====

P2 executou por 1 u.t.
P2 vai para a fila de Impressão.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P2

===== INSTANTE 20 =====

P4 executou por 1 u.t.
P2 executou 1 u.t. do seu I/O de Impressão, faltam 3 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P2

===== INSTANTE 21 =====

P4 executou por 1 u.t.
P4 sofreu preempção, vai pro final da fila de baixa prioridade.
P2 executou 1 u.t. do seu I/O de Impressão, faltam 2 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P1 P3 P0 P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P2

```



```

===== INSTANTE 22 =====

P1 executou por 1 u.t.
P2 executou 1 u.t. do seu I/O de Impressão, faltam 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P1 P3 P0 P4
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão: P2

===== INSTANTE 23 =====

P1 executou por 1 u.t.
P1 sofreu preempção, vai pro final da fila de baixa prioridade.
P2 executou 1 u.t. do seu I/O de Impressão, faltam 0 u.t.
P2 finalizou seu I/O de Impressão, vai pra fila de alta prioridade.

Fila de alta prioridade: P2
Fila de baixa prioridade: P3 P0 P4 P1
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 24 =====

P2 executou por 1 u.t.

Fila de alta prioridade: P2
Fila de baixa prioridade: P3 P0 P4 P1
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 25 =====

P2 executou por 1 u.t.
P2 sofreu preempção, vai pra fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P3 P0 P4 P1 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 26 =====

P3 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P3 P0 P4 P1 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 27 =====

P3 executou por 1 u.t.
P3 sofreu preempção, vai pro final da fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P0 P4 P1 P2 P3
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

```

===== INSTANTE 28 =====

P0 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P0 P4 P1 P2 P3
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 29 =====

P0 executou por 1 u.t.

P0 sofreu preempção, vai pro final da fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P2 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 30 =====

P4 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P4 P1 P2 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 31 =====

P4 executou por 1 u.t.

P4 finalizou.

Fila de alta prioridade:
Fila de baixa prioridade: P1 P2 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 32 =====

P1 executou por 1 u.t.

P1 finalizou.

Fila de alta prioridade:
Fila de baixa prioridade: P2 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 33 =====

P2 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P2 P3 P0
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

```

===== INSTANTE 34 =====

P2 executou por 1 u.t.
P2 sofreu preempção, vai pro final da fila de baixa prioridade.

Fila de alta prioridade:
Fila de baixa prioridade: P3 P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 35 =====

P3 executou por 1 u.t.

Fila de alta prioridade:
Fila de baixa prioridade: P3 P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 36 =====

P3 executou por 1 u.t.
P3 finalizou.

Fila de alta prioridade:
Fila de baixa prioridade: P0 P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 37 =====

P0 executou por 1 u.t.
P0 finalizou.

Fila de alta prioridade:
Fila de baixa prioridade: P2
Fila de I/O Disco:
Fila de I/O Fita:
Fila de I/O Impressão:

===== INSTANTE 38 =====

P2 executou por 1 u.t.
P2 finalizou.

Nenhuma fila com processos, CPU ociosa.

Não há mais processos, escalonamento finalizado com sucesso.

===== FINAL =====
Tempos de Turnaround:

TT (P0): 34 u.t
TT (P1): 29 u.t
TT (P2): 34 u.t
TT (P3): 34 u.t
TT (P4): 31 u.t

```