

Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação



MAC0438 - Programação Concorrente
EP3 - Filósofos Famintos
Professor: Daniel Macedo Batista

Autoras:
Bárbara de Castro Fernandes - 7577351
Taís Aparecida Pereira Pinheiro - 7580421

São Paulo - SP, 21 de junho de 2015

Sumário

1	Introdução	3
2	Desenvolvimento	4
2.1	Bibliotecas	4
2.2	Estruturas principais	4
2.3	Implementação do Monitor	5
2.3.1	Variáveis locais	5
2.3.2	Métodos	5
2.4	Função Principal	5
2.4.1	Execução do Programa em modo Uniforme	5
2.4.2	Execução do Programa em modo com Pesos	5
3	Dificuldades	5
4	Modo de execução	6
4.1	Compilação	6
4.2	Entrada	6
5	Referências	7

1 Introdução

O problema pode ser resumido como alguns filósofos sentados ao redor de uma mesa redonda, cada qual fazendo exclusivamente uma das duas coisas: **comendo** ou **pensando**.

Existe um único prato na mesa com R porções de comida. Um garfo é posicionado entre cada par adjacente de filósofos (portanto, cada filósofo tem exatamente um garfo à sua esquerda e exatamente um garfo à sua direita). Assume-se que um filósofo necessita de dois garfos para poder comer e além disso, um filósofo só pode utilizar um garfo que esteja imediatamente à sua esquerda ou imediatamente à sua direita.

2 Desenvolvimento

2.1 Bibliotecas

Das bibliotecas da linguagem C utilizadas neste EP, destacam-se graças às suas funcionalidades as seguintes:

- **pthread.h**: Responsável pela criação e gerenciamento das *threads* que representam cada filósofo.
- **semaphore.h**: Define um tipo de variável **semáforo**, e é utilizada para realizar as operações com semáforos necessárias para o funcionamento correto das *threads* e implementação do módulo **Monitor**.

2.2 Estruturas principais

As estruturas principais utilizadas na elaboração deste exercício-programa são estruturas já prontas localizadas nas três bibliotecas citadas anteriormente.

Filosofo

```
typedef struct filosofo
{
    int id;
    int peso;
    int refeicoesComidas;
    Estado estado;
} Filosofo;
```

Estado

```
typedef enum estado {PENSANDO, COMENDO, FAMINTO} Estado;
```

A estrutura **Filosofo** retém as características de cada Filósofo em particular, como o seu estado, id, peso e número de refeições já feitas.

2.3 Implementação do Monitor

Como a linguagem em que foi implementada este programa é **C** e não oferece suporte para uso de Monitores nativo da linguagem, o monitor utilizado foi implementado em um módulo a parte nomeado como **monitor** e utilizado através da biblioteca **monitor.h**.

2.3.1 Variáveis locais

Não condicionais: `sem_t`, `semGarfos`, `semRefeicoes` e `semImpressao`;
Condicionais: `semFilosofos`

2.3.2 Métodos

- **void inicializaMonitor(int,int,Filosofo*)**: Responsável pela inicialização do monitor;
- **void wait(condicao*)**: Função encarregada da função **wait** do monitor;
- **void signal(condicao*)**: Função encarregada da função **signal** do monitor;
- **void ficaFaminto(int, Filosofo*)**: Responsável por colocar o um Filósofo em estado *Faminto* e verificar se ele consegue tomar os garfos que lhe são possíveis.
- **void ficaFaminto(int, Filosofo*)**: Responsável por colocar o um Filósofo em estado *Faminto* e verificar se ele consegue tomar os garfos.
- **void tentaPegarGarfos(int, Filosofo*)**: Complementar ao método *ficaFaminto*, verificar se um filósofo consegue tomar os garfos que lhe são possíveis.
- **void soltaGarfos(int i,Filosofo*, int*)**: Responsável por após um filósofo comer uma porção de comida os garfos serem devolvidos.

2.4 Função Principal

2.4.1 Execução do Programa em modo Uniforme

Para execução do ciclo do programa em **modo Uniforme** designamos um único método:
void *filosofoModoUniforme(void *param): Responsável por permitir que os filósofos pensem e comam sem diferenciação.

2.4.2 Execução do Programa em modo com Pesos

3 Dificuldades

As dificuldades que podemos destacar neste programa foram: Falta de suporte da linguagem em relação à implementação de monitores e conciliação desta tarefa com as demais de fim de semestre. Por estes motivos o modo de execução com Pesos não foi concluído.

4 Modo de execução

Este projeto foi entregue em um arquivo comprimido de extensão **.tar.gz** contendo 5 arquivos:

- **ep3.c**: Arquivo com o código fonte deste exercício-programa.
- **monitor.c**: Arquivo com o código fonte desta implementação do monitor.
- **monitor.h**: Arquivo header do código fonte monitor.c.
- **Makefile**: Makefile para compilação do programa.
- **LEIAME**: Arquivo com a descrição técnica no programa.
- **Relatório.pdf**: Este relatório.

4.1 Compilação

Para compilar, digite:

```
$ make
```

4.2 Entrada

O programa principal pode ser executado da seguinte maneira:

- **./ep3 R U|P**

5 Referências

Referências

- [1] Enunciado do exercício-programa
- [2] Análise do problema