

# MAC0438 – Programação concorrente – 1s2015

## EP3

Data de entrega: 22/6/2015

Prof. Daniel Macêdo Batista

## 1 Problema

Neste EP você deverá resolver o problema dos *filósofos famintos*.

$N > 2$  filósofos estão sentados em uma mesa circular. Cada filósofo tem dois possíveis estados na sua vida: pensando e comendo. Para que um filósofo possa comer ele precisa pegar os dois garfos que estão imediatamente à sua esquerda e à sua direita, sendo que na mesa há exatamente  $N$  garfos, cada um colocado entre cada par de filósofos. Cada um dos filósofos tem ainda um peso  $p_i (i \in 1, \dots, N) > 1$  e gasta um tempo aleatório pensando. Todos os filósofos gastam o mesmo tempo para comer quando eles conseguem obter os dois garfos necessários para isso.

Simule o problema dos filósofos famintos utilizando monitor, sendo que cada filósofo deve ser uma thread. Além de seguir a descrição acima, a sua implementação deve garantir a justiça evitando que algum filósofo morra de fome em duas situações:

- Situação uniforme: ao observar o jantar dos filósofos por um longo intervalo de tempo, as quantidades de porções (considere que a cada vez que um filósofo consegue pegar os dois garfos, ele come 1 porção) comidas por cada filósofo devem estar bem próximas umas das outras;
- Situação com peso: ao observar o jantar dos filósofos por um longo intervalo de tempo, as quantidades de porções comidas por cada filósofo devem ser diretamente proporcionais aos pesos deles. Filósofos mais pesados devem comer mais vezes do que filósofos mais leves.

**Obs.: Sua solução não deve inserir restrições a mais no problema.**

## 2 Requisitos

### 2.1 Monitor

Neste EP você deverá manipular monitores utilizando **apenas** as funções vistas em sala de aula com exatamente os mesmos nomes abaixo:

```
empty(cv), wait(cv), wait(cv,rank), signal(cv), signal_all(cv), minrank(cv)
```

Note que seu EP não precisa utilizar todas as funções (obviamente as funções `signal` e `wait` são obrigatórias). EPs que não implementem pelo menos as funções `signal` e `wait` receberão nota ZERO. Você deve indicar no LEIAME do seu EP onde estão as implementações de todas as funções que você implementou para manipular monitores.

**A descrição do monitor que você implementar deve ser explicada no LEIAME do seu código.**

## 2.2 Linguagem

Seu programa pode ser escrito em qualquer linguagem de programação desde que o compilador seja gratuito e tenha versão disponível para GNU/Linux.

Algumas linguagens já possuem algumas estruturas voltadas para a implementação de programas com monitores. Leia a página da wikipedia a respeito disso:

[https://en.wikipedia.org/wiki/Monitor\\_\(synchronization\)#History](https://en.wikipedia.org/wiki/Monitor_(synchronization)#History)

Caso você utilize alguma linguagem que já venha com as funções exigidas na subseção anterior implementadas, você pode utilizá-las, mas note que os nomes exigidos para as funções **devem** ser mantidos. Por exemplo, se em uma dada linguagem a função para acordar um processo da fila de uma variável de condição é `SignalZ`, você precisa criar um *alias*, uma outra função, etc... chamada `signal` que por sua vez chamará a `SignalZ`.

## 2.3 Entrada e Saída

O seu EP deverá receber na linha de comando os seguintes parâmetros nesta ordem:

arquivo R U|P

arquivo é o caminho para um arquivo texto no seguinte formato:

N

p1 p2 p3 p4 p5 p6 ... pN

onde  $N$  e  $p_i$  são as variáveis inteiras apresentadas no início deste documento.

R é a quantidade de porções comidas que o EP deve simular. Ou seja, se você definir R como sendo 1, assim que 1 filósofo comer, o seu programa deve terminar. Se R fosse 2, o seu programa terminaria após duas porções serem comidas pelos filósofos (não há problema se nesse caso por exemplo um mesmo filósofo venha a comer as 2 porções). Seu programa deve terminar quando exatamente R porções forem comidas. Nenhuma porção a mais pode ser comida.

Quando a opção U for passada na linha de comando, o EP deve executar a situação uniforme, e quando a opção for P ele deve executar a situação com peso, ambas explicadas no início deste documento.

O seu EP deverá imprimir na saída padrão as seguintes informações toda vez que um filósofo conseguir obter os dois garfos para comer:

- Identificador do filósofo que obteve os garfos;
- Instante de tempo que o filósofo começou a comer.

Quando um filósofo terminar de comer, o seu EP deverá imprimir na saída padrão:

- Identificador do filósofo que terminou de comer;
- Instante de tempo que o filósofo terminou de comer.

No fim da execução o seu EP também deverá imprimir na saída padrão:

- Quantidade de vezes que cada filósofo comeu;

### 3 Sobre a entrega

Você deverá submeter no paca um arquivo .tar.gz contendo os seguintes itens:

- fonte(s), Makefile (ou similar), arquivo LEIAME e um relatório em .pdf.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep3-membros\_da\_equipe. Por exemplo: ep3-joao-maria.

O relatório deve relatar o funcionamento do seu programa em quatro casos:

1. Entrada simples (poucos filósofos) na situação U
2. Entrada simples (poucos filósofos) na situação P
3. Entrada complexa (muitos filósofos) na situação U
4. Entrada complexa (muitos filósofos) na situação P

Note que dois arquivos devem ser usados. Um para os casos 1 e 2 e outro para os casos 3 e 4. O conteúdo do arquivo que você utilizou deve ser apresentado no relatório. Os valores dos parâmetros ficam ao seu critério. Note que o R precisa ser alto o suficiente para que os padrões esperados para a saída do programa apareçam.

Em cada caso você deve apresentar um gráfico e escrever um texto explicando esse gráfico, comentando se os resultados foram esperados e, se não foram os esperados, porquê. O gráfico deve ser em barras, onde o eixo x deve ser o identificador de cada filósofo e o eixo y deve ser a quantidade de porções que ele comeu, ou seja, a informação impressa pelo programa na última porção comida. É de se esperar que no caso U as barras tenham os mesmos tamanhos e que no caso P as barras tenham tamanhos proporcionais aos pesos de cada filósofo.

O EP pode ser feito individualmente ou em dupla.