

# Relatório EP4 - SO turma 94

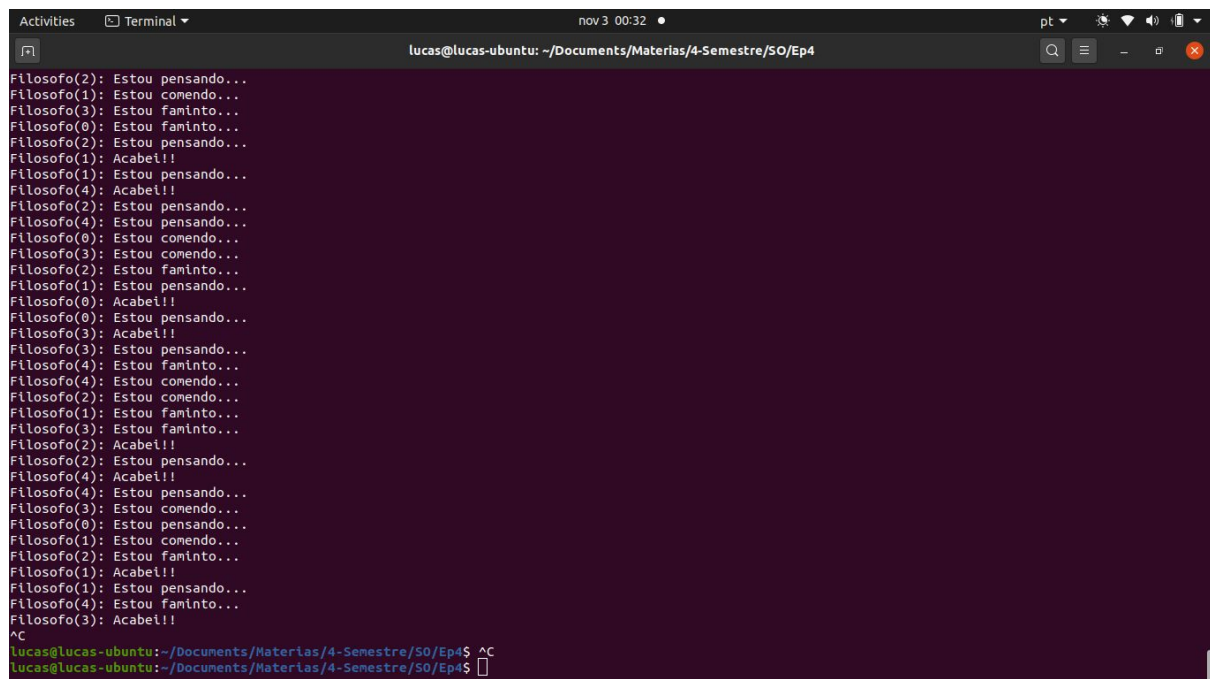
Lucas Mendes Sales , NUSP: 11270736

Utilizei como base o código do Ep3, onde criava cinco threads que concorriam pelo acesso a uma variável, que era controlado por um semáforo da *lib semaphore* do *pthread*.

Busquei representar os filósofos como threads e os hashis como semáforos binários, já que estes últimos teriam dois estados possíveis (disponível ou indisponível), logo, foram criados 5 semáforos e 5 threads. O vetor global “hashis”, foi usado para guardar os ponteiros dos semáforos. A intenção era que o filósofo esperasse para pegar o primeiro hashi (up no semáforo), o segundo, e de posse dos dois, comesse, em seguida soltasse ambos (down) e isso é o que foi feito na função *acao*, que é executada dentro de um *loop* dentro na função *vivendo*, sendo esta a função passada no momento da criação das threads.

Como esperado (no livro texto é citado algo parecido), essa primeira parte da solução gerou um *deadlock*, e, para eliminá-lo, foi criado o vetor *estados*, que guarda os estados dos filósofos como p(pensando), f(faminto) ou c(comendo), para que seja feito um teste da situação dos filósofos vizinhos antes do filósofo em questão tentar pegar os hashis para comer (executar a função ação), então, se ambos vizinhos estiverem famintos, ele não tenta pegar os hashis (a função ação não é executada). Com isso tudo pareceu funcionar.

Exemplos de saída:



```
Activities  Terminal  nov 3 00:32  pt  lucas@lucas-ubuntu: ~/Documents/Materias/4-Semestre/SO/Ep4
Filosofo(2): Estou pensando...
Filosofo(1): Estou comendo...
Filosofo(3): Estou faminto...
Filosofo(0): Estou faminto...
Filosofo(2): Estou pensando...
Filosofo(1): Acabei!!
Filosofo(1): Estou pensando...
Filosofo(4): Acabei!!
Filosofo(2): Estou pensando...
Filosofo(4): Estou faminto...
Filosofo(0): Estou comendo...
Filosofo(3): Estou comendo...
Filosofo(2): Estou faminto...
Filosofo(1): Estou pensando...
Filosofo(0): Acabei!!
Filosofo(0): Estou pensando...
Filosofo(3): Acabei!!
Filosofo(3): Estou pensando...
Filosofo(4): Estou faminto...
Filosofo(4): Estou comendo...
Filosofo(2): Estou comendo...
Filosofo(1): Estou faminto...
Filosofo(3): Estou faminto...
Filosofo(2): Acabei!!
Filosofo(2): Estou pensando...
Filosofo(4): Acabei!!
Filosofo(4): Estou pensando...
Filosofo(3): Estou comendo...
Filosofo(0): Estou pensando...
Filosofo(1): Estou comendo...
Filosofo(2): Estou faminto...
Filosofo(1): Acabei!!
Filosofo(1): Estou pensando...
Filosofo(4): Estou faminto...
Filosofo(3): Acabei!!
^C
lucas@lucas-ubuntu:~/Documents/Materias/4-Semestre/SO/Ep4$ ^C
lucas@lucas-ubuntu:~/Documents/Materias/4-Semestre/SO/Ep4$
```

```
Activities Terminal nov 3 00:45 pt
lucas@lucas-ubuntu: ~/Documents/Materias/4-Semestre/SO/Ep4

[c - p - c - p - p]
[c - f - c - p - p]
[c - f - c - f - p]
[p - f - c - f - p]
[p - f - c - f - p]
[p - f - p - f - p]
[p - c - p - f - p]
[p - c - p - c - p]
[p - c - p - c - p]
[p - c - p - c - p]
[p - p - p - p - c]
[p - c - p - c - p]
[p - p - f - p - c]
[p - p - c - p - c]
[p - c - p - c - c]
[f - p - c - p - c]
[f - p - c - p - c]
[f - p - p - f - c]
[f - p - p - c - c]
[f - p - p - c - p]
[c - p - c - p - p]
[c - f - p - c - p]
[c - f - p - c - p]
[c - f - p - c - p]
[p - c - p - p - p]
[p - c - p - p - c]
[p - f - p - p - p]
[p - c - p - p - c]
[p - p - p - f - c]
[p - p - p - f - p]
[f - p - p - c - p]
[c - p - p - c - p]
^C
lucas@lucas-ubuntu:~/Documents/Materias/4-Semestre/SO/Ep4$ ^C
lucas@lucas-ubuntu:~/Documents/Materias/4-Semestre/SO/Ep4$
```

Aqui vê-se a o vetor que guarda os estados dos filósofos

Experimento feito individualmente no SO Ubuntu 20.04.1 LTS instalado na máquina.

Compilador:

gcc version 9.2.1 20191008 (Ubuntu 9.2.1-9ubuntu2)

Comando para compilacao: gcc v1.c -lpthread -o v1

Execucao: ./v1

### Fontes:

Livro texto: Abrahan Silberschatz, Peter Baer Galvin, Greg Gagne, Fundamentos de Sistemas Operacionais 9ª ed.

[https://www.inf.pucrs.br/~emoreno/undergraduate/CC/sisop/class\\_files/Aula09b.pdf](https://www.inf.pucrs.br/~emoreno/undergraduate/CC/sisop/class_files/Aula09b.pdf)