



Escola de Artes, Ciências e Humanidades
da Universidade de São Paulo

EDGAR HENRIQUE DE OLIVEIRA LIRA (12717266)
PEDRO AUGUSTO DE MELO DELAMURA SOARES (12542800)

SISTEMAS OPERACIONAIS

EP4

SÃO PAULO
2022

EDGAR HENRIQUE DE OLIVEIRA LIRA (12717266)
PEDRO AUGUSTO DE MELO DELAMURA SOARES (12542800)

SISTEMAS OPERACIONAIS

EP4

Trabalho apresentado à Escola de Artes,
Ciências e Humanidades da Universidade de
São Paulo como requisito da disciplina de
Sistemas Operacionais.

Prof. Dr.^a Gisele da Silva Craveiro.

SÃO PAULO
2022

SUMÁRIO

1 CÓDIGO	3
1.1 Código	3
1.2 Prints da tela	4
1.3 Ambiente utilizado	5
2 SEMÁFOROS UTILIZADOS	6

1 CÓDIGO

1.1 Código

```
#include<stdio.h>
#include<stdlib.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>

sem_t prato;
sem_t garfo[5];

typedef struct {

char *nome;
int numero;
} filosof;

void* filosofo(void *pensador) {
    filosof* fil=(filosof *)pensador;
    sleep(3);

    while(1){

        sleep(1);
        sem_wait(&prato);
        sem_wait(&garfo[fil->numero]);
        printf("%s pegou o garfo %d\n",fil->nome , fil->numero);
        sem_wait(&garfo[(fil->numero+1)%5]);
        printf("%s pegou o garfo %d\n",fil->nome, (fil->numero+1)%5);
        printf("%s esta comendo.\n",fil->nome);
        sleep (1);
        printf("%s soltou os garfos %d e %d\n", fil->nome, fil->numero,
(fil->numero+1)%5);

        sem_post(&garfo[(fil->numero+1)%5]);
        sem_post(&garfo[fil->numero]);
        printf("%s voltou a pensar...\n",fil->nome );
        sem_post(&prato);

    }
}
```

```

int main(){

    filosof pensador[5];
    pthread_t tid[5];

    printf("-----Iniciando o Jantar dos Filósofos-----\n\n");

    pensador[0].nome = "Sócrates";
    pensador[1].nome = "Platao";
    pensador[2].nome = "Aristoteles";
    pensador[3].nome = "Democrito";
    pensador[4].nome = "Espinoza";

    sem_init(&prato,0,2);

    for(int i=0;i<5;i++){
        sem_init(&garfo[i],0,1);
    }

    for(int i=0;i<5;i++){
        pensador[i].numero = i;
        pthread_create(&tid[i],NULL,filosofo,(void *)&pensador[i]);
        printf("%s esta pensando...\n", pensador[i].nome);
    }

    for(int i=0;i<5;i++)
        pthread_join(tid[i],NULL);
}

```

1.2 Print da saída, como o código foi compilado e executado.

```
[edgarlira@Arch EP4]$ ./jantar
-----Iniciando o Jantar dos Filósofos-----

Sócrates esta pensando...
Platao esta pensando...
Aristoteles esta pensando...
Democrito esta pensando...
Espinoza esta pensando...
Platao pegou o garfo 1
Platao pegou o garfo 2
Platao esta comendo.
Sócrates pegou o garfo 0
Platao soltou os garfos 1 e 2
Platao voltou a pensar...
Sócrates pegou o garfo 1
Sócrates esta comendo.
Aristoteles pegou o garfo 2
Aristoteles pegou o garfo 3
Aristoteles esta comendo.
Sócrates soltou os garfos 0 e 1
Sócrates voltou a pensar...
Aristoteles soltou os garfos 2 e 3
Aristoteles voltou a pensar...
Espinoza pegou o garfo 4
Espinoza pegou o garfo 0
Espinoza esta comendo.
Democrito pegou o garfo 3
Espinoza soltou os garfos 4 e 0
Espinoza voltou a pensar...
Democrito pegou o garfo 4
Democrito esta comendo.
Platao pegou o garfo 1
Platao pegou o garfo 2
Platao esta comendo.
Democrito soltou os garfos 3 e 4
Democrito voltou a pensar...
Platao soltou os garfos 1 e 2
Platao voltou a pensar...
```

1. 3 Ambientes utilizados

Ambiente usado para programar: Vscode

Compilação: gcc jantar.c -o jantar -lpthread

Execução: ./jantar

Ambiente usado para compilar e executar: gnome-terminal

2 Semáforos utilizados

Para a solução do problema do jantar dos filósofos, apresentada previamente, foram utilizados dois tipos de semáforos, isto é, o semáforo binário e o semáforo de contagem.

Primeiramente, foi considerado que existe um único “prato” na mesa, e que, uma vez que existem cinco filósofos, somente dois podem comer simultaneamente (o piso da divisão de 5 por 2). Assim, foi utilizado um semáforo de contagem para permitir que somente dois filósofos “acessem” o prato (no máximo 4 filósofos poderiam ultrapassar o semáforo, mas optamos por apenas 2 nesta solução), pois caso os cinco pudessem acessar, haveria o risco de cada filósofo pegar o garfo à sua esquerda ou direita simultaneamente e provocar um deadlock. Dessa forma, mesmo no caso em que dois filósofos que estejam lado a lado acessem o prato ao mesmo tempo, um deles pegará dois garfos, e o outro irá esperar a liberação do outro garfo, e, após comer, o filósofo libera os dois garfos.

Já para os garfos, foram utilizados cinco semáforos binários, uma vez que existem cinco garfos. O uso desse tipo de semáforo é justificado pois cada garfo possui somente dois estados possíveis: está sendo usado ou não está sendo usado. Dessa forma, um filósofo só pode pegar um garfo que não esteja sendo usado, fechando o semáforo para aquele garfo, e abrindo o semáforo para aquele garfo quando acabar de comer e soltá-lo.