Animação Multithread com Semáforos

MC504 - Sistemas Operacionais

Áureo Henrique

Jonas Cardoso

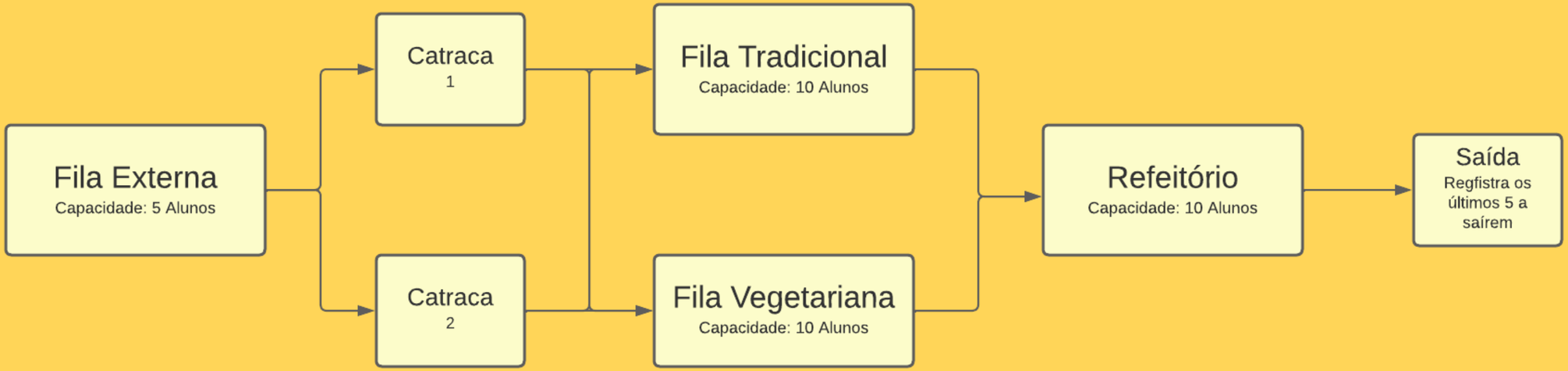
Lethycia Maia

Samuel souza

A ideia do projeto é desenvolver uma animação para demonstrar como funciona o restaurante universitário fazendo uso de multithreads com semáforo

Mecânica do Restaurante

- Entrar na fila externa e esperar
- Passar pela catraca
- Entrar na fila interna e esperar
- Ir para as mesas comer
- Sair do restaurante



Uso das threads

- Cria-se uma thread para cada aluno que vai comer
- Cada thread percorre toda a função f_estudante e passa por todas etapas do restaurante universitário, desde a entrada até a saída

Etapa - Fila externa

- Cada thread dos alunos entra na fila externa e aguarda os estudantes a sua frente para passar pela catraca

Etapa - Catraca

- Ao entrar na catraca, o estado dela muda para ocupado até que o estudante saia dela e vá para a fila interna, seja ela a tradicional ou a vegetariana

Etapa - Filas internas

- Agora na fila interna, o estudante aguarda sua vez para retirar sua refeição e ir em direção ao refeitório

Etapa - Refeitório

- Ao entrar no refeitório, o estudante tem um determinado tempo de refeição pré-determinado, que ao finalizar o mesmo vai a saída

Etapa - Sair

- Finalizando as etapas, a saída é a ultima parte do processo, ao chegar nela o estudante finaliza o percurso e finaliza o processo da thread

Visualização

```
                    BANDECO
/-----------------------------------------------------\    Entrada
<  16   17   18   19   20   21   22   23   24   25  L
|    ^ Fila Tradicional ^ - v Fila Vegetariana v    |   36   37   38   39   40
<  26   27   28   29   30   31   32   33   34   35  L
|-----------------------------------------------------|
>                  v Refeitório v                   >
| 15   15   13   12   11   10    9    8    7    6   |    5    4    3    2    1
> ==   ==   ==   ==   ==   ==   ==   ==   ==   ==   >
\-----------------------------------------------------/    Saída
```

Função imprime Cineminha

```c
int alunosFilaExterna[N_CAPACIDADE_FILA_EXTERNA];
int alunosFilaTradicional[N_CAPACIDADE_FILA_INTERNA];
int alunosFilaVegetariana[N_CAPACIDADE_FILA_INTERNA];
int alunosCatracas[N_CATRACAS];
int alunosAssentos[N_ASSENTOS];
int alunos[N_ALUNOS];

/* Funcao responsavel pela impressao do cineminha. */
void imprimeCineminha() {
  int i, qtdAlunosFE = 0, qtdAlunosFT = 0, qtdAlunosFV = 0, qtdAlunosCO = 0, qtdAlunosEX = 0, titulo =
0;
  int alunosFE[N_ALUNOS], alunosFT[N_ALUNOS], alunosFV[N_ALUNOS], alunosCO[N_ALUNOS],
alunosEX[N_ALUNOS];

  /* Inicialzia parametros das filas. */
  for (i = 0; i < N_ALUNOS; i++){
    alunosFE[i] = INT_MAX;
    alunosFT[i] = INT_MAX;
    alunosFV[i] = INT_MAX;
    alunosCO[i] = INT_MAX;
    alunosEX[i] = INT_MAX;
  }
 [...]
 }
```

```c
int main() {
  fVeg = criar_lista();
  fTrad = criar_lista();

  pthread_t thr_alunos[N_ALUNOS];
  int id_alunos[N_ALUNOS];

  sem_init(&sem_catraca, 0, N_CATRACAS);
  sem_init(&sem_assento, 0, N_ASSENTOS);
  sem_init(&sem_estados, 0, 1);
  sem_init(&sem_verifFilaExterna, 0, N_CAPACIDADE_FILA_EXTERNA);
  sem_init(&sem_estado_catracas, 0, 1);


  for (int i = 0; i < N_FILA_INTERNA; i++){
    sem_init(&sem_filaInterna[i], 0, N_CAPACIDADE_FILA_INTERNA);
    sem_init(&sem_verifFilaInterna[i], 0, 1);
  }

  destruir_lista(fVeg);
  destruir_lista(fTrad);
  return 0;
}
```

Inicialização dos semáforos

```
if (sem_trywait(&sem_filaInterna[minha_fila]) == 0) {
    // code
}
```

Try Wait

Execução - Saída Inicial

```
                        BANDECO
/──────────────────────────────────────────────────────────\    Entrada
<  ..   ..   ..   ..   ..   ..   ..   ..   ..   ..   L
|    ^ Fila Tradicional ^ - v Fila Vegetariana v     |    1    2    3    4    5
<  ..   ..   ..   ..   ..   ..   ..   ..   ..   ..   L
|──────────────────────────────────────────────────────────|
>                    v Refeitório v                    >
| ..   ..   ..   ..   ..   ..   ..   ..   ..   ..   |    ..   ..   ..   ..   ..
> ==   ==   ==   ==   ==   ==   ==   ==   ==   ==   >
\──────────────────────────────────────────────────────────/    Saída
```

Execução - Saída intermediária

```
                        BANDECO
/──────────────────────────────────────────────────────────\    Entrada
<  21   22   23   24   25   26   27   28   ..   ..  0
|    ^ Fila Tradicional ^ - v Fila Vegetariana v    |   37   38   39   40   41
<  29   30   31   32   33   34   35   ..   ..   ..  L
|──────────────────────────────────────────────────────────|
>                    v Refeitório v                    >
| 20   19   18   17   16   15   14   13   12   11   |   10    9    8    7    6
> ==   ==   ==   ==   ==   ==   ==   ==   ==   ==   >
\──────────────────────────────────────────────────────────/    Saída
```

Execução - Saída Final

```
                    BANDECO
/──────────────────────────────────────────────────────\    Entrada
<  ..   ..   ..   ..   ..   ..   ..   ..   ..   ..  L
|    ^ Fila Tradicional ^ - v Fila Vegetariana v    |    ..   ..   ..   ..   ..
<  ..   ..   ..   ..   ..   ..   ..   ..   ..   ..  L
|──────────────────────────────────────────────────────|
>                  v Refeitório v                    >
| ..   ..   ..   ..   ..   ..   ..   ..   ..   ..   |    45   44   43   42   41
> ==   ==   ==   ==   ==   ==   ==   ==   ==   ==   >
\──────────────────────────────────────────────────────/    Saída
```

Valeu!