Banheiro Unisex com 3 Gêneros

Gustavo Henrique Grützmann (19102053) Vitor Sorpile Geraldo (19102063)

I. INTRODUÇÃO

Este projeto tem como objetivo apresentar uma solução para o problema do Banheiro Unisex com 3 Genêros. Foi desenvolvida uma solução genérica para um banheiro com N boxes e que divide o tempo de espera para usar o banheiro de forma justa entre os gêneros, através do cálculo do tempo médio de espera das pessoas de cada gênero. Para resolver os três problemas apresentandos na atividade (1 box para 60 pessoas igualmente divididas entre os gêneros, 150 pessoas para 3 boxes e 300 pessoas para 5 boxes) foi usado o mesmo código, apenas alterando os parâmetros que indicam o número total de pessoas e se é para gerar o mesmo número de pessoas para cada gênero na classe que gera as pessoas e o parâmetro que indica o número de pessoas na classe que representa o banheiro. Ao final do programa é impresso o tempo médio de espera de cada gênero para usar o banheiro, a taxa de ocupação de cada box e a quantidade de pessoas de cada gênero que usaram o banheiro.

II. DIAGRAMAS

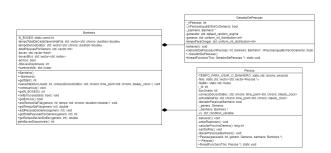


Fig. 1. Diagrama de Classes

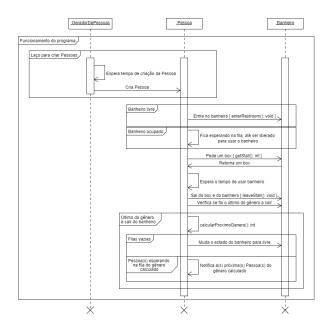


Fig. 2. Diagrama de Sequência

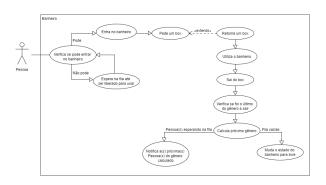


Fig. 3. Diagrama de Casos de Uso

III. CLASSES

A. Banheiro

Classe responsável por representar um banheiro, faz o controle de acesso ao banheiro e aos boxes através de *mutexes*, possui variáveis para contar o tempo de uso de cada de box, o tempo que cada gênero ficou na fila esperando e o número total de pessoas de cada gênero, assim como funções para acessar essas variáveis e suas duas principais funções: *getStall* que reserva e retorna o primeiro box disponível no banheiro e *leaveStall* que libera o box recebido por parâmetro e a partir do *time_point*, também recebido por parâmetro, que representa o começo do uso do box, é calculado por quanto tempo tal box foi utilizado.

B. GeradorDePessoas

Classe derivada de *std::thread* responsável por gerar as pessoas que vão usar o banheiro, recebe no seu construtor o número de pessoas a serem geradas e se é para gerar o mesmo número de pessoas para os 3 gêneros.

C. Pessoa

Classe derivada de *std::thread* responsável por representar uma pessoa, possui variáveis para armazenar quando entrou na fila, quando começou a usar o box, qual foi o box utilizado, se está liberado para usar o banheiro, seu gênero e seu id. Também possui variáveis *static* para que as próprias pessoas consigam acessar as filas (existe uma fila para cada gênero) para quando sair do banheiro, notificar as pessoas do próximo gênero a usar o banheiro.

Assim que uma pessoa é criada, é passado para o construtor de *std::thread* a função *threadFunction*, que apenas chama o método *behavior*, nesse a pessoa tenta entrar no banheiro através da função *enterRestroom*, se ele não estiver sendo utilizado, consegue entrar e continua a execução, caso contrário fica esperando através de uma variável condicional até ser notificada. Após isso, pede ao banheiro um box pela função *getStall* e recebe o número do box disponível, então usa o banheiro por 5 segundos e sai dele usando o método *leaveStall*. Caso a pessoa seja a última a sair entre as pessoas do mesmo gênero que estavam usando o banheiro, ela calcula o tempo médio de espera dos gêneros que possuem pessoas esperando e notifica o máximo de pessoas do gênero com maior tempo médio de espera que puderem usar o banheiro ao mesmo tempo.

IV. COMPILAÇÃO E EXECUÇÃO

O programa é compilado com um *Makefile* e pode ser executado através do comando "./banheiro_problemaN", onde N é o número do problema desejado, para uma execução simples ou através de "make valgrind" para uma execução com a avaliação de erros de memória pelo *valgrind*.

V. TESTES

A fim de verificar o correto funcionamento do programa, foram executados vários testes com 1, 3 e 5 boxes e diferentes números de pessoas a usar o banheiro. As flags utilizadas na compilação foram -Wall -Wextra -Wshadow -Wconversion -Wcast-align -Werror -std=c++17 -pthread -pedantic -g e para verificar se existiam erros de memória o programa foi executado com o valgrind com as flags -leak-check=full - track-origins=yes -s.

```
stimerent judicitife annexi. Anti-Carl Ambeiro, miss. on a, Jonesed size and pictor annexi. Section 1: 1. Increase distriction production of the pictor of t
```

Fig. 4. 2 execuções com 1 box e 20 pessoas de cada gênero

```
Separation of the control of the con
```

Fig. 5. 2 execuções com 3 boxes e 50 pessoas de cada gênero

```
And the control of th
```

Fig. 6. 2 execuções com 3 boxes e 150 pessoas

```
And the second s
```

Fig. 7. 2 execuções com 5 boxes e 100 pessoas de cada gênero

```
And the state of t
```

Fig. 8. 2 execuções com 5 boxes e 300 pessoas