

Projeto 2
Estruturas de Dados, Turma E, 1/2014
Prof. Dúbio

Uma simulação controlada por eventos é um programa de computador que mimetiza o comportamento de pessoas ou objetos em um sistema, em resposta a eventos que ocorrem em certos instantes de tempo. O programa deve manter uma estrutura de dados para cada pessoa ou objeto, e colocá-lo em uma fila de acordo com o tempo e/ou prioridade do evento. O objetivo deste projeto é escrever um programa de simulação que gerencia o uso das pistas de um grande aeroporto. As pistas são usadas para pousos e decolagens e os requisitos e parâmetros desejados são descritos a seguir.

O aeroporto possui três (3) pistas, 1, 2 e 3, sendo que a pista 3 somente é usada para decolagens, a menos que ocorra uma emergência. As pistas 1 e 2 são usadas tanto para pousos quanto para decolagens. A cada 1 hora ($UnTempo=5min$), de 0 a 42 (Nvoos) voos requerem pousos ou decolagens ao aeroporto. Os voos são identificados por duas (2) letras maiúsculas, seguidos por 4 números inteiros. Uma lista com 42 códigos de voos é fornecida. A letra A, indicando Aproximação, seguido de um valor inteiro de 0 a 12, onde o valor 0 indica prioridade máxima de pouso (0 combustível, ou o intervalo máximo temporal que o mesmo pode aguardar (cada desses números indicam um intervalo de 5 min), ou a letra D, de Decolagem (sem valores marcados de combustível). Após a autorização na fila para o voo em aproximação aterrissar o mesmo passa a entrar na fila de Pouso (letra P). Os voos devem ser atendidos em uma estratégia de fila, ou quando uma emergência (e.g. 0 combustível para qualquer Aproximação, ou 3 Aproximações ao mesmo tempo) acontecer, sendo que estes então terão prioridades. Se ocorrer de 4 ou mais aeronaves se aproximarem com 0 nível de combustível uma mensagem de “ALERTA GERAL DE DESVIO DE AERONAVE” deve ser emitido, alocando 3 e indicando quais serão desviadas mas sem necessitar indicar outro aeroporto para essas. Um relógio global deve ser inicializado e toda a lista gerada aleatoriamente no início da execução deve ser tratada até o final. O programa de simulação deverá gerar aleatoriamente (ver função exemplo) os valores das seguintes variáveis: NVoos (0 a 42), NAproximações (0 a 26), NDecolagens (0 a 16), CombA (0 a 12). Sendo que $NAproximações + NDecolagens = NVoos$. Após gerar os valores, o programa deve selecionar os códigos dos voos da lista, e montar a fila de atendimento. A simulação deverá cobrir todos os pedidos gerados, um a um, e imprimir a sequência de atendimento na tela. Considerar os tempos médios (L, G e A) como fixos a cada simulação. Para o projeto essas podem ser variáveis globais, ou geradas aleatoriamente também no início da execução.

Elementos a serem mostrados na Tela típica de saída:

“Aeroporto Internacional de Paraíso”

Hora Atual:

Fila de Pedidos:

NVoos:

Naproximações:

NDecolagens:

Filas Designadas (na ordem) por Pistas:

Pista 1(sequência e relógio com previsões de atendimento na fila (A, P, D)):

Pista 2(sequência e relógio com previsões de atendimento na fila (A, P, D)):

Pista 3(sequência e relógio com previsões de atendimento na fila (A, P, D)):

Exemplo de lista de sequências de códigos de voos para o programa: [“VG3001”, “JJ4404”, “LN7001”, “TG1501”, “GL7602”, “AZ1009”, “VG3002”, “RL7880”, “AL0012”, “TT4544”, “TG1505”, “VG3003”, “JJ4403”, “JJ4401”, “LN7002”, “AZ1002”, “AZ1007”, “GL7604”, “AZ1003”, “JJ4403”, “TG1502”, “GL7601”, “TT4500”, “RL7801”, “GF4681”, “GF4690”, “AZ1020”, “JJ4435”, “AZ1065”, “LF0978”, “RL7867”, “GL7675”, “LF0972”, “VG3070”, “GL7672”, “AZ1079”, “VG3072”, “RL7870”, “AL0072”, “TT4574”, “TG1575”, “VG3073”]

Função exemplo para gerar números aleatórios:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
```

```
int geraRand(int min, int max)
{
    int r;
    srand(time(NULL));
    r=min+rand()%(max-min+1);
    return r;
}
```

Escreva um programa em linguagem C, o qual deverá simular o controle de prioridade de pousos e decolagens no aeroporto (parâmetros como fornecidos). A prioridade deve ser implementada como fila, e os parâmetros gerados aleatoriamente a cada chamada do programa.

O código deve ser bem documentado, de forma modular com funções para cada tarefa independente, realizado por dois (2) estudantes do curso usando “*pair programming*”, e entregue via sistema <http://aprender.unb.br> do curso, no prazo estipulado.