



Trabalho Prático 4

Filas – Simulação de Atendimento de Clientes em Agências Bancárias

Valor: 0,5 pts (5% da nota total)

Documentação não-Latex: -0,1 pts

Interface gráfica com simulador: +0,1 pts

Impressão Frente-Verso: +0,05 pts (ecologicamente correto)

Data de entrega:

26/05/2009

O objetivo deste trabalho é aplicar o conceito de filas computacionais para simular o atendimento das filas de clientes em caixas de agências bancárias.

A agência é composta basicamente de três entidades: cliente (que fazem fila para ser atendidos); caixa (o local físico); e atendente de caixa (a pessoa que atende).

Para elaborar o simulador, considere as seguintes regras.

- A agência bancária abre as 10 horas da manhã e fecha as 16 horas. Portanto, a agência fica aberta por 6h durante o dia.
Defina uma macro (`#define`) para este valor, e.g., `TEMPSIM`;
- Um cliente não deve ficar mais que 30 minutos na fila (resolução do código do consumidor). Defina uma macro para este valor, e.g. `TEMPCLI`;
- O tempo de atendimento de cada cliente deve variar de 1 minutos até 10 minutos.
Defina macros (`#define`) para estes valores, e.g., `TEMPCLIMIN` e `TEMPCLIMAX`;
- Devem existir duas filas: uma preferencial (para idosos, aposentados, gestantes, deficientes físicos, etc.) e outra normal (para os clientes não preferencial);
- Os atendentes devem sempre dar prioridade a cliente preferencial. Todavia após o atendimento consecutivo de dois clientes preferencial, o caixa deve atender um cliente não preferencial. Exceção: caso a fila de clientes não preferencial esteja vazia, o atendente pode continuar prestando serviços aos clientes preferencial.
- Inicialmente, quando a agência abre, somente um caixa e um atendente estarão trabalhando;
- Para finalizar, quando a agência fechar, nenhum cliente mais entra na agência (e consequentemente nas filas).
Porém os clientes que ainda estão dentro da agência devem ser atendidos.
- Caixas podem ser ativados e desativados caso o “gerente” considere necessário;
- Considere que as quantidades mínima e máxima de caixas sejam definidas por macros, e.g. `QTDCAxmin` e `QTDCAxmax`;
- Um atendente de caixa não pode ficar trabalhando direto (sem intervalos) mais de 3h.
Defina uma macro (`#define`) para este valor, e.g., `TEMPATDmax`;
- Um atendente de caixa deve ter um intervalo de descanso/almoço de pelo menos 30 minutos. Defina uma macro (`#define`) para este valor, e.g., `TEMPATDint`;
- Considere que as quantidades mínima e máxima de atendentes sejam definidas por macros,



.e.g. QTDATDMIN e AQTDTDMAX;

- O simulador deve ter resolução de 1 minuto, ou seja, os controles devem ser atualizados minuto a minuto: um cliente pode chegar, pode sair da fila; uma caixa pode abrir, pode fechar, etc.

Ao final do expediente (quando a agência bancária fechar e não existir mais clientes para serem atendidos), devem ser impressos: a quantidade de clientes atendidos por cada caixa e cada atendente, o tempo médio de atendimento de cada caixa e cada atendente e dos clientes; a quantidade de clientes que não foram atendidos dentro de 30 minutos (TEMPCLI).

Procure simular o cenário da agência com variações que contemplem o amplo espectro do problema.

Dicas:

Defina macros para determinar os eventos de forma aleatória, como por exemplo: a quantidade máxima de clientes que chega por minuto à agência, e então utilize números aleatórios (rand()) para determinar a quantidade de clientes que chega em um dado minuto.

O que deve ser entregue

- Código fonte do programa em C ou C++ (bem identada e comentada).
- Documentação do trabalho. Entre outras coisas, a documentação deve conter:
 1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.
 3. Estudo de Complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados e do programa como um todo (notação O).
 4. Listagem de testes executados: os testes executados devem ser simplesmente apresentados.
 5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso
 7. Em Latex: Caso o trabalho não seja elaborado/escrito em latex, perde-se 0,1 pontos.
 8. Impressão: Caso o trabalho seja impresso usando os dois lados da folha (impressão frente-verso), ganha-se 0,05 pontos extra – um incentivo à “preservar o planeta”.
 9. Formato: mandatoriamente em PDF (<http://www.pdf995.com/>).



Universidade Federal de Ouro Preto – UFOP
Instituto de Ciências Exatas e Biológicas – ICEB
Departamento de Computação – DECOM
Disciplina: Algoritmos e Estrutura de Dados I – CIC102 / 20091
Professor: David Menotti (menottid@gmail.com)



Obs1: Consulte as dicas do Prof. Nívio Ziviani de como deve ser feita uma boa implementação e documentação de um trabalho prático:
<http://www.dcc.ufmg.br/~nivio/cursos/aed2/roteiro/>

Obs2: Veja modelo de como fazer o trabalho em latex:
<http://www.decom.ufop.br/prof/menotti/aedI091/tps/modelo.zip>
(caso alguém desenvolva um modelo similar em word, favor me enviar)

Como deve ser feita a entrega:

A entrega DEVE ser feita pelo Moodle na forma de um **único** arquivo zipado, contendo o código, os arquivos e a documentação. Também deve ser entregue a documentação impressa na próxima aula (teórica ou prática) após a data de entrega do trabalho.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, identificação e comentários no programa também serão avaliados;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero;
- Trabalhos entregue em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.