UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA
AUTOMAÇÃO EM TEMPO REAL
PROFESSOR: Constantino Seixas Filho

PROVA 2 (com consulta)

1) [30 pontos] Complete o programa abaixo que mostra o uso de memória compartilhada em Delphi.

```
var
  HMapping: THandle;
  PMapData: Pointer;

const
  MAPFILESIZE = 1000;

procedure OpenMap;
var
  llInit: Boolean;
  lInt: Integer;
begin
  HMapping := CreateFileMapping(_____________, nil, PAGE_READWRITE,
                0, MAPFILESIZE, pchar('NomeEscolhido'));
  // testa se já existe
  llInit := (GetLastError() <> ERROR_ALREADY_EXISTS);
  if (hMapping = 0) then
  begin
    ShowMessage('Não consegue criar Mapa');
    Application.Terminate;
    exit;
  end;
  PMapData := ____________________________________________;
  if PMapData = nil then
  begin
    CloseHandle(HMapping);
    ShowMessage('Não consegue visualizar Mapa');
    Application.Terminate;
    exit;
  end;
  if (llInit) then
  begin
    // Inicializa bloco para #0 se recém criado
    memset(PMapData, #0, MAPFILESIZE);
  end
end;

procedure CloseMap;
begin
  if PMapData <> nil then
    _____________________________________;
  if HMapping <> 0 then
    CloseHandle(HMapping);
end;

var
  HMapMutex: THandle;

const
  REQUEST TIMEOUT = 1000;

function LockMap:Boolean;
begin
  Result := true;
  HMapMutex := CreateMutex(_____________________________);
  if HMixMutex = 0 then
  begin
    ShowMessage('Não consigo criar Mutex');
    Result := false;
  end
  else
  begin
```

```
    if                                          = WAIT_FAILED then
       begin
      // timeout
      ShowMessage('Não consigo realizar lock');
      Result := false;
    end
  end
end;


procedure UnlockMap;
begin
                          ;
  CloseHandle(HMixMutex);
end;
```

2) [20 pontos] Programação em tempo real

Você tem quatro tarefas com as seguintes necessidades de processamento:

Tarefa	Duração (ms)	Periodicidade (ms)
T1	10	100
T2	40	100
T3	40	400
T4	30	100

- Organize estas tarefas num mecanismo de agenda síncrona com tamanho de página de 100 ms.
- Este sistema seria factível utilizando um escalonador assíncrono do tipo EDF ?
- Mostre o diagrama de escalonamento para escalonador EDF.

Suponha que além disso uma interrupção de duração de 15 ms pode acontecer a qualquer instante. Depois de ocorrer ela seguramente não ocorre pelos próximos 200 ms. O sistema continua factível ? Demonstre.

3) [30 pontos] O problema do banheiro unissex

Suponha que no seu departamento na Universidade existe apenas um banheiro, que pode ser visitado por homens e mulheres, mas não simultaneamente. Quando o banheiro estiver vazio, tanto um homem quanto uma mulher podem entrar. Se houver um homem no banheiro outros cavalheiros poderão entrar e quando houver uma mulher no banheiro outras continuarão a ser admitidas. Quando a última mulher sair deverá dar prioridade a um homem e vice versa. O banheiro comporta no máximo quatro pessoas de cada vez.

Propriedades de segurança:
- Não pode haver pessoas de sexos diferentes no banheiro.
- Não pode haver *deadlock*.
- O algoritmo não precisa ser justo.

Use mensagens assíncronas do Windows para resolver o problema. Todas as threads homens e as threads mulheres devem realizar uma solicitação para entrar no banheiro através de uma mensagem enviada a uma thread administradora, e depois entrar na fila do semáforo Brasileiros ou do semáforo Brasileiras (em homenagem ao Sarney). A thread administradora autoriza a entrada das pessoas a serem admitidas. Quando sai do banheiro cada thread avisa ao administrador através de uma mensagem enviada ao mesmo administrador.

4) [20 pontos] Como você usaria *Completion Ports* para resolver o seguinte problema: você deseja enviar valores inteiros indicando uma operação a ser realizada por um *pool* de threads (*task farm*). Cada thread, após resolver o problema envia a resposta, um valor *double* para a thread mestre. O mecanismo deve determinar a correspondência entre dado e resultado.