

**SSC0501 - Introdução à Ciência da Computação I (Teórica)**

**Professor responsável:** *Fernando Santos Osório*

**Semestre:** 2011/1

**Horário:** Seg. 21h-22h40 - Terça 19h-20h40

**E-mail:** fosorio @ icmc.usp.br

fosorio @ gmail.com

**Web:** <http://www.icmc.usp.br/~fosorio/>

*Nome do Aluno:* \_\_\_\_\_

**Número USP :** \_\_\_\_\_

**DATA:** 13 / 06 / 2011

***PROVA TEÓRICA – P2 Final (Prova Escrita)***

**Implemente um programa modular e bem estruturado de uma agenda para controle de compromissos, que deve possuir sub-rotinas que realizam as seguintes tarefas: [avaliação geral: 2.0 pts]**

**0. Inicialização do programa (leitura do arquivo de configuração) [1.5 pts]**

**1. Cadastrar compromissos na agenda [2.0 pts]**

**2. Salvar em disco os compromissos cadastrados [1.5 pts]**

**3. Entrar no modo de “monitoramento de compromissos” [3.0 pts]**

**Descrição geral do programa:** O programa não deve possuir nenhuma variável global, podendo ser aceita apenas uma exceção, a da declaração do vetor de dados que armazena o cadastro de compromissos, onde somente esta variável pode ser global. O programa deve começar sua execução lendo automaticamente um arquivo de configuração denominado de “**config.txt**” (implementado em uma sub-rotina). Após lido o arquivo de configuração, o programa deve apresentar um menu com 3 opções para o usuário, cada uma implementada por um sub-rotina: **opção 1:** cadastrar um novo compromisso na agenda (cadastra um evento apenas por vez); **opção 2:** salvar todos os compromissos agendados em um arquivo em disco (“compromissos.txt”); **opção 3:** monitorar os compromissos, avisando quando chegar a hora de um determinado evento. O programa poderá usar algumas rotinas adicionais disponibilizadas ao programador, e para tanto deve usar o arquivo “extras.h”, que contém os cabeçalhos das rotinas extras descritas abaixo:

Rotinas disponíveis:

*/\* Rotina que consulta a data atual \*/*

**void Le\_data (int \*dia, int \*mes, int \*ano, char \*dia-da-semana);**

*/\* Rotina que consulta a hora atual.*

**void Le\_hora (int \*hora, int \*min);**

*/\* Rotina liga alarme sonoro. A variável tipo\_alarme contém ‘N’ se normal e ‘U’ se urgente \*/*

**void Alarme\_on (char tipo\_alarme);**

*/\* Rotina desliga alarme sonoro \*/*

**void Alarme\_off (void);**

*/\* Rotina que verifica se a tecla ESC está sendo pressionada, faz o teste e segue adiante, “não para” (retorna 1 se estiver pressionada, 0 no caso contrário) \*/*

**int Pressionou\_ESC (void);**

*/\* Rotina que envia uma mensagem de e-mail avisando sobre a hora de um compromisso atual \*/*

**void Envia\_email (char \*destinatario, char \*mensagem, int dia, int mes, int ano, int hora, int min);**

Exemplo da tela de entrada de dados do programa:

```
<< Agenda de Compromissos ACME Ver:0.01 >>
>> Leitura do Arquivo de Configuração <<
OK

>> Menu:
1 – Cadastrar Compromisso
2 – Salvar Cadastro de Compromissos
3 – Monitorar Compromissos

Digite sua opção: 1
>> Cadastrar Compromisso <<
Data – Dia, Mes,Ano: 13 06 2011
Horario – Hora, Min : 21 00
Compromisso: Prova_P2_SSC0501
Tipo do Compromisso: urgente

Digite sua opção: 1
>> Cadastrar Compromisso <<
Data – Dia, Mes,Ano: 14 06 2011
Horario – Hora, Min : 23 00
Compromisso: Comemorar_Final_Aulas_SSC0501
Tipo do Compromisso: normal

Digite sua opção: 2
>> Salvar Compromissos <<
OK

Digite sua opção: 3
>> Monitorando Compromissos <<
  [Digite ESC para voltar ao menu]
13/06/2011 - 21:00 Alarme Ativo(U): Prova_P2_SSC0501
13/06/2011 - 21:01 Alarme Desativado
                                     (... Programa Executando ... Teclou ESC!)

Digite sua opção: _
```

**Etapa 0. Inicialização do programa e leitura do arquivo de configuração:** O arquivo de configuração é um arquivo texto que contém 3 linhas, onde na primeira linha está indicado o tamanho máximo da agenda de compromissos (nro. máximo de compromissos que podem ser agendados), na segunda linha do arquivo está indicado quantos minutos antes do compromisso o alarme deve ser ativado (valor 0, 5 ou 10, indicando na hora, com 5 minutos ou 10 minutos antes do compromisso), e na terceira linha está indicado o endereço de e-mail para onde deve ser enviado o aviso de compromisso, podendo ser um endereço de e-mail válido ou então o texto “nomsg” que significa que não devem ser enviadas mensagens automáticas de e-mail avisando sobre um compromisso. Exemplo do arquivo de configuração “config.txt”

**100**

**0**

**fulano@gmail.com**

**(ou “nomsg” se não desejar o envio de e-mails)**

**Etapa 1. Cadastrar compromissos na agenda:** O usuário ao selecionar esta opção deverá fornecer a data (dia, mês e ano) e o horário (hora, minutos) de um determinado compromisso, além de indicar um texto que descreve este compromisso e por fim indicar se o compromisso é do tipo “urgente” ou “normal”. Estes dados devem ser cadastrados em um vetor de compromissos contendo uma estrutura de dados adequada para o registro dos compromissos. A sub-rotina de cadastrar compromissos é chamada a partir do menu, a cada vez que o usuário desejar incluir um novo compromisso no cadastro (é cadastrado apenas 1 compromisso por vez).

**Etapa 2. Salvar em disco os compromissos cadastrados:** Esta opção permite que seja gerado um arquivo texto em disco com toda a lista (cadastro) dos compromissos agendados. O arquivo deve possuir o seguinte formato de saída: na primeira linha do arquivo será gravada a data no formato (dd/mm/aaaa), na segunda linha será gravada a hora no formato (hh:mm), na terceira linha será gravado o texto que descreve o compromisso, e na quarta linha será gravado um indicativo se o compromisso é “normal” (gravar ‘N’) ou se é “urgente” (gravar ‘U’), repetindo assim para todos os demais compromissos. Veja abaixo um exemplo do “compromissos.txt” arquivo a ser gerado:

```
13/06/2011
21:00
Prova_P2_SSC0501
U
14/06/2011
23:00
Comemorar_Final_Aulas_SSC0501
N
```

**Etapa 3. Entrar no modo de “monitoramento de compromissos”:** Esta opção permite que o usuário ative o modo de monitoramento de compromissos, onde o programa irá ficar testando constantemente a data e horário atual e verificando se tem algum compromisso a ser executado na presente data e horário. Portanto, ao selecionar esta opção o programa irá ficar “preso” repetindo a verificação para determinar se deve tocar o alarme avisando a chegada do horário de algum compromisso. Esta rotina deve executar diversas tarefas:

1. Note que para sair desta opção, o usuário deverá pressionar a tecla ESC, e o programa deve constantemente testar para ver se esta tecla foi pressionada (rotina extra), e caso tenha sido pressionada, deve retornar ao menu principal.
2. Enquanto aguarda a chegada do horário de um compromisso, o programa deve ficar verificando a data e hora atual (através das rotinas extras de leitura de data e hora), e caso haja um compromisso agendado dentro do horário atual (considerando o horário do compromisso agendado), deve avisar sobre este compromisso;
3. Uma vez chegada a hora de um compromisso, o programa deverá então ativar o alarme sonoro usando a rotina para este fim (rotina extra), sempre lembrando de ativar o tipo de alarme correto (urgente ou normal) para cada compromisso.
4. O programa também deve escrever na tela uma mensagem indicando que foi ativado o alarme, com a data e hora de ativação (ver exemplo da tela apresentado).
5. Uma vez ativado o alarme, o programa também deve considerar o envio de uma mensagem de e-mail alertando que está no horário de um compromisso, e enviando assim um aviso para o destinatário definido no arquivo de configuração (se o destinatário for “nomsg” então não deverão ser enviadas mensagens por e-mail).
6. Passado o horário de ativação do alarme de compromisso, o programa deve desativar o alarme (rotina extra), escrever uma mensagem na tela avisando que desativou o alarme, e prosseguir sempre monitorando novos alarmes de compromissos.

**Desafio: [0.5 pts do item “monitoramento de compromissos”]**

O monitoramento de compromissos, para ser considerado 100% completo deverá considerar também o tempo de aviso prévio. O aviso prévio, quando for diferente de zero (conforme definido no arquivo de configuração), deve ser considerado para que o alarme seja ativado com uma certa antecedência (5 ou 10 minutos) antes do horário programado para o compromisso.

**Observações Finais:**

- ⇒ Esta prova possui 4 etapas mas se constitui de um **programa COMPLETO** que deve incluir estas etapas, implementadas de modo modular com uso de sub-rotinas, com a adequada passagem de parâmetros, sem globais (exceto o vetor principal do programa), e sem desvios do tipo “GoTo”.
- ⇒ *O desrespeito a estas regras implicará em desconto na nota, mas mesmo assim a prova ainda será avaliada em relação ao seu conteúdo restante (este quesito não é eliminatório).*

---

---

**REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA**

1. A PROVA É **INDIVIDUAL**.
2. A PROVA É **COM CONSULTA AO MATERIAL INDIVIDUAL**.  
(Pode consultar: cadernos, anotações, livros – qualquer tipo de material escrito ou impresso)
3. **NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL** (Cadernos, Anotações, Livros, etc).
4. **NÃO É PERMITIDO O USO DE DISPOSITIVOS ELETRÔNICOS** durante a prova.  
(Não pode usar: notebook, computador, palmtops/pdas, celular, etc.)
5. RESPONDER A PROVA NAS FOLHAS FORNECIDAS: A CANETA OU A LÁPIS.  
**SE FOR RESPONDIDA A LÁPIS E TIVER QUALQUER INDÍCIO DE ALTERAÇÃO OU RASURA, NÃO SERÃO ACEITOS PEDIDOS DE REVISÃO DE PROVA.**
6. LEMBRE-SE DE **IDENTIFICAR A PROVA** COM O SEU NOME E NÚMERO USP.  
DEVOLVER A FOLHA DE RESPOSTAS JUNTAMENTE COM A PROVA (Questões).
7. DURAÇÃO: 3 Horas

>> *Não será tolerado qualquer tipo de troca de informações entre alunos durante a prova!*

>> *Responda a prova de forma LEGÍVEL. Se não for possível entender o texto, é questão errada!*

>> *Qualquer dúvida, chame o professor e aguarde SENTADO em sua cadeira para ser atendido.*

>> *Somente se dirija ao professor quando for entregar a prova.*

---

---

FIM

---

---