tp2- Trabalho prático de CI1001
Introdução à passagem de parâmetros por endereços
Agenda simples, versão 2
Departamento de Informática/UFPR
André Grégio, Luis Bona e Marcos Castilho

1 Sobre a entrega do trabalho

São requisitos para atribuição de notas a este trabalho:

- Uso de um arquivo makefile para facilitar a compilação. Os professores rodarão "make" e deverão obter o arquivo executável funcional com a sua solução. Este executável, cujo nome deverá ser tp2, deverá estar no subdiretório tp2;
- Ao compilar, incluir pelo menos `-Wall -Werror -Wextra`. Se não compilar, o trabalho vale zero. Haverá desconto por cada *warning*;
- Arquivo de entrega:
  - Deve estar no formato tar comprimido (.tar.gz);
  - O tar.gz deve ser criado considerando-se que existe um diretório com o nome do trabalho. Por exemplo, este trabalho é o tp2;
  - Então seu tar.gz deve ser criado assim:
    - Estando no diretório tp2, faça:
    - `cd ..`
    - tar zcvf tp2.tar.gz tp2)
  - Desta maneira, quando os professores abrirer o tar.gz (com o comando tar zxvf tp2.tar.gz) terão garantidamente o diretório correto da entrega para poderem fazer a correção semi-automática.
  - O que colocar no tar.gz? Todos os arquivos que são necessários para a compilação, por isso se você usa arquivos além dos especificados, coloque-os também. Mas minimamente deve conter todos os arquivos `.c`, `.h` e o makefile;
  - Os professores testarão seus programas em uma máquina do departamento de informática (por exemplo, cpu1), por isso, antes de entregar seu trabalho faça um teste em máquinas do dinf para garantir que tudo funcione bem.

2 O trabalho

Você deve baixar o tp2.tar.gz anexo a este enunciado e abrí-lo para poder fazer o trabalho, pois irá precisar de todos os arquivos ali contidos:

libAgenda.h: arquivo (read only) de *header* com todos os protótipos das funções para manipular a agenda;

makefile: sugestão de um makefile que você pode usar (ou adaptar, se quiser).

O arquivo `.h` não pode ser alterado. Na correção, os professores usarão os arquivos `.h` originais.

- Use boas práticas de programação, como indentação, bons nomes para variáveis, comentários no código, bibliotecas, defines... Um trabalho que não tenha sido implementado com boas práticas vale zero.
- Quaisquer dúvidas com relação a este enunciado devem ser solucionadas via email para `prog1prof@inf.ufpr.br` pois assim todos os professores receberão os questionamentos. Na dúvida, não tome decisões sobre a especificação, pergunte!
- Dúvidas podem e devem ser resolvidas durante as aulas.

3 O problema

Este trabalho é uma variante de implementação do mesmo problema do TP1, o desafio é aprender a passar parâmetros usando ponteiros (endereços) para evitar cópias de estruturas que podem ser complexas e ocupar muito espaço na *STACK*.

Isso é muito importante para o restante do curso, por isso aprenda o conceito e os detalhes semânticos e sintáticos o quanto antes!

Basicamente, modificamos o arquivo `libAgenda.h` e as modificações devem ser adaptadas na sua solução do TP1.

4 Exemplo de entrada e saída

A sua entrada e saída deverá ser exatamente como especificado abaixo, senão o trabalho valerá zero!

- Note que não há mensagens para entrada de dados, apenas saída.
- Entre com os dados exatamente assim:
  - o ano em uma única linha;
  - dia, mês, ano e data, nesta ordem, na mesma linha, separados por um único espaço em branco;
  - um char.
- O laço termina quando o char "s" for lido;
- Após o laço, imprimir agenda conforme abaixo, mas somente se ela tiver algum compromisso, senão não imprima nada. Use com estes comandos printf assim (adaptando os nomes das suas variáveis, evidentemente):

```
printf ("dia: %3d, ", sua_variavel_para_o_dia_entre_0_e_364);
printf ("ano: %4d, ", obtemAno (sua_variavel_agenda));
printf ("hora: %2d, compromisso!\n", sua_variavel_hora_entre_1_e_24);
```

```
2023
1 1 2023 12
Compromisso inserido com sucesso!
c
31 12 2023 23
Compromisso inserido com sucesso!
c
1 1 2023 12
Data/Hora ocupada, compromisso nao inserido
c
50 1 2023 1
Data e/ou hora invalidos, compromisso nao inserido
c
1 50 2023 1
Data e/ou hora invalidos, compromisso nao inserido
c
Data e/ou hora invalidos, compromisso nao inserido
c
1 1 2023 50
Data e/ou hora invalidos, compromisso nao inserido
s
```

```
dia:   0, ano: 2023, hora: 12, compromisso!
dia: 364, ano: 2023, hora: 23, compromisso!
```

5 Considerações finais

Se você implementou corretamente o TP1 não deverá ter dificuldade nesta nova implementação, a estrutura principal dos códigos não muda muito, tudo é uma questão de você entender o conceito de parâmetros que usam ponteiros (endereços) e aprender a sintaxe correspondente no código.

Bom trabalho!