tp1- Trabalho prático de CI1001
Introdução a manipulação de estruturas
Agenda simples
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 deverá estar no subdiretório tp1;
- Opções de compilação: deve incluir pelo menos `-Wall`, `Werror` e `Wextra`. Haverá desconto na nota se o compilador mostrar algum erro ou *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 tp1;
  - Então seu tar.gz deve ser criado no diretório pai do subdiretório tp1, o qual deve conter todos os arquivos que serão entregues (tar zcvf tp1.tar.gz tp1), de maneira que os professores, ao abrir o tar.gz com o comando "tar zxvf tp1.tar.gz" obterão um diretório tp1 com as suas respostas;
  - 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 tp1.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;

libAgenda.c: arquivo de implementação com uma das funções do .h já implementada. Você deve implementar as demais;

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, ... Um trabalho que não tenha sido implementado com boas práticas vale zero.
- Quaisquer dúvidas com relação a este enunciado devem ser solucionadas via Moodle sendo que todos os professores devem receber o questionamento. Na dúvida, não tome decisões sobre a especificação, pergunte!
- Dúvidas podem e devem ser resolvidas durante as aulas.

3 O problema

Para entender estruturas, você deve criá-las e manipulá-las. Estruturas são muito importantes em C para a criação de Tipos Abstratos de Dados, isto é, representações abstratas de objetos reais para uso e processamento via sistemas computacionais.

Neste trabalho, o desafio é criar uma agenda que represente a agenda física: um "caderno" que abrange o intervalo de um ano e possui espaços para cada dia deste ano, dividido em horas.

A abstração a ser feita é bem simples, pois nesta agenda, um compromisso é somente uma hora marcada como ocupada (identificada pelo inteiro 1). Entretanto, há várias estruturas pré-definidas que se encadeiam para formar a agenda. A manipulação de tais estruturas por meio das funções disponibilizadas no *header* dado é a chave para uma implementação bem sucedida do trabalho.

No programa principal, deve-se:

1. Criar uma agenda para o ano;
2. Prover um laço para o usuário que permite, até que o caracter 's' (de 'sair') seja pressionado:

- Ler uma data e hora de um compromisso;
- Validar a data e hora lidas do usuário dentro do intervalo possível de datas e dentro do ano estabelecido para a criação da agenda;
- Verificar na agenda se data e hora estão livres;
- Marcar um compromisso se data/hora estiverem disponíveis;
- Avisar o usuário se data/hora estiverem indisponíveis.

3. Após montar a agenda do ano, deve-se listar todos os compromissos marcados de maneira organizada (um por linha, com data e hora).

4 Considerações finais

Esse trabalho vai oferecer a oportunidade de amadurecimento na Linguagem C e em especial no uso de "structs". Comentários adicionais sobre as funções a serem implementadas encontram-se no arquivo de *header*.

Bom trabalho!