

Professore: Fernando V. Paulovich (paulovic at icmc.usp.br)
Aluno PAE: Grabiél Dias Cantareira (gabrielcantareira at gmail.com)

Trabalho 01

Sistema de Gerenciamento de Cadastro

1 Prazos, Especificações e Observações importantes:

- O trabalho descrito a seguir é individual e não será tolerado qualquer tipo de plágio ou cópia em partes ou totalidade do código. Caso seja detectada alguma irregularidade, os envolvidos serão chamados para conversar com o professor responsável pela disciplina e os trabalhos serão zerados.
- A entrega deverá ser feita única e exclusivamente por meio do *Sistema Run Codes* no endereço eletrônico <https://run.codes/> até o dia **15 de Setembro de 2016 as 23 horas e 59 minutos**. Sejam responsáveis com o prazo final para entrega, o *Run Codes* está programado para não aceitar submissões após este horário e não será aceito entrega fora do sistema.
- A interpretação desta descrição faz parte do trabalho. Leia a descrição do trabalho com atenção e várias vezes, anotando os pontos principais e as possíveis formas de resolver o problema. Comece a trabalhar o quanto antes para não ficar dúvidas e você consiga entregar o trabalho a tempo.
- O trabalho deverá ser submetido em formato zip/Makefile contendo o arquivo principal, o Makefile e possíveis TAD's implementadas (pares .h .c). Atente para o fato que todos os arquivos enviados deverão conter um cabeçalho contendo nome e número USP.
- A implementação é livre, crie quantos TAD's julgarem necessárias. Importante notar que a modularização do código bem como a forma como as TAD's foram criadas serão levados em consideração na atribuição final da nota.
- Compile o programa em um sistema operacional linux antes de submeter ao *Run Codes*. **Trabalho em que o comando make e make run não funcione, não será corrigido e consequentemente receberá nota zero.** Caso não tenha o sistema operacional linux instalado em sua maquina, aconselha-se a utilização de uma máquina virtual. Como instalar o sistema operacional linux em uma maquina virtual está descrito no seguinte tutorial: <http://maistutoriais.com/2012/01/instalar-maquina-virtual-usando-o-virtual-box/>.
- Referencie, com um comentário no próprio código, qualquer algoritmo ou trecho de código retirado da internet. Código copiado sem a devida referencia é considerado plágio que por sua vez é crime.
- Nenhuma saída do trabalho será em arquivo, ou seja, o resultado dos algoritmos deverá ser exibidos utilizando o stdout `printf`.

2 Descrição do Problema:

Implemente um sistema de gerenciamento de cadastro capaz de guardar o primeiro nome, o ultimo nome, o cpf e o número do telefone do usuário. Uma particularidade do sistema é que todo o armazenamento deve ser feito em uma lista circular duplamente ligada com nó cabeça (sentinela).

3 Comandos do Sistema:

- ‘sair’ - Este comando deverá sair do sistema liberando toda memória possivelmente alocada;
- ‘cadastrar’ - Este comando deverá cadastrar um novo usuário com todas as suas informações. Nenhuma informação irá conter espaço em branco entre suas palavras. Os dados de cada usuário serão informados na seguinte ordem:

1. ‘Primeiro Nome’ - máximo 15 caracteres.
2. ‘Ultimo Nome’ - máximo 15 caracteres.
3. ‘CPF’ - máximo 15 caracteres.
4. ‘Telefone’ - máximo 20 caracteres.

- ‘buscar *cpf*’ - Este comando deverá buscar os dados do usuário de acordo com o CPF. Os dados do usuário devem ser impressos na tela da seguinte maneira:

```
printf("Primeiro Nome: %s\n", primeiroNome);  
printf("Ultimo Nome: %s\n", ultimoNome);  
printf("CPF: %s\n", cpf);  
printf("Telefone: %s\n", telefone);  
printf("-----\n");
```

- ‘dump’ - Este comando deverá imprimir na tela os dados de todos os usuários na ordem com que eles foram inseridos no sistema. Estes dados devem ser impressos da mesma maneira como é feito no comando de busca.
- ‘remover *cpf*’ - Este comando deverá invalidar os dados do usuário de acordo com o CPF. Ao invalidar estes dados, o usuário não mais deverá aparecer na saída dos comandos dump e buscar.

4 Sugestão de Makefile

```
all: clean item.o lista.o locadora.o main.o compile clean

compile:
    @gcc lista.o locadora.o main.o -o trab1 -lm

main.o:
    @gcc -c main.c

locadora.o:
    @gcc -c locadora.c

lista.o:
    @gcc -c lista.c

item.o:
    @gcc -c item.c

clean:
    @find -name '*' | xargs rm -rf {}
    @find -name '*.o' | xargs rm -rf {}

run:
    @./trab1

zip:
    zip -r trab1 *.c *.h Makefile
```

5 Sugestão do esqueleto do main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "locadora.h"

int main(int argc, char *argv[]) {

    char *command = (char *)malloc(32*sizeof(char));
    int sair = 0;

    do{
        scanf("%s",command);

        if(strcmp(command,"sair") == 0) {
            sair = 1;
        } else if(strcmp(command,"cadastrar") == 0) {
            scanf("%s",...); //primeiro nome
            scanf("%s",...); //ultimo nome
            scanf("%s",...); //cpf
            scanf("%s",...); //telefone
            cadastrar(...);
        } else if(strcmp(command,"buscar") == 0) {
            scanf("%s",...); //cpf
            buscar(...);
        } else if(strcmp(command,"remover") == 0) {
            scanf("%s",...); //cpf
            remover(...);
        } else if(strcmp(command,"dump") == 0) {
            dump();
        }
    } while(!sair);

    free(command);

    return 0;
}
```