

A universidade está comprometida em preservar a reputação do seu diploma. Para garantir que cada diploma outorgado possua o significado que ele realmente representa, nesta disciplina exige-se que o acadêmico:

- Entregue um trabalho original, isto é, desenvolvido de forma integral e individual pelo próprio acadêmico.
- Não utilize uma solução parcial ou completa obtida de qualquer outra fonte (outra pessoa, Internet, etc).
- Não discuta seu trabalho com outros colegas.
- Proteja seu trabalho e evite que ele seja roubado. Não compartilhe seu trabalho com ninguém.
- Não entregue um trabalho que não é seu.

O não cumprimento destas regras caracteriza desonestade acadêmica e a sua nota pode ser zerada. Além disso, penalidades como suspensão ou expulsão podem ser impostas pelo departamento de Engenharia de Computação.

Nesta atividade você demonstrará domínio básico sobre a linguagem C: arrays, ponteiros, alocação de memória dinâmica, estruturas, funções, passagem de parâmetros, operações com bits, diretivas do pré-compilador, manipulação de arquivos, entre outros.

O programa a ser desenvolvido simula um conjunto restrito de operações que são normalmente realizadas por sistemas embarcados de Televisão Digital. Ao ser executado, o programa deve apresentar o menu abaixo para permitir o usuário interagir com o sistema. O programa só deve terminar quando o usuário escolher a opção zero.

```
Por favor selecione uma das opções abaixo:
0 - Sair
1 - Exibir programação (listar canais)
2 - Adicionar novo canal na lista
3 - Remover canal da lista
4 - Exibir apenas os canais bloqueados
5 - Procurar canal (número)
6 - Procurar canal (nome)
7 - Gravar lista no disco (ASCII)
8 - Carregar lista do disco (ASCII)
9 - Gravar lista no disco (binário)
10 - Carregar lista do disco (binário)
```

Um canal é composto das informações a seguir e que podem ser armazenadas em 1156 bytes:

- **Bloqueado** (1 bit): valor numérico que armazena 1 se o canal estiver bloqueado e 0 se estiver liberado para exibição.
- **Número** (15 bits): valor inteiro para armazenar o número do canal.
- **Preço** (16 bits): valor inteiro com o custo para desbloquear o canal e permitir sua exibição.
- **Nome** (1024 bits): *string* terminada com caractere nulo. Indica o nome da emissora.
- **Programa** (8192 bits): *string* terminada com caractere nulo. Indica o nome do programa em exibição.

Sua implementação deve utilizar a estrutura abaixo para armazenar as informações pertinentes a um canal:

```
typedef struct Canal {
    char bloqueado;
    short int num;
    short int preco;
    char nome[128];
    char programa[1024];
} canal_t;
```

Uma estrutura deste tipo poderia ser inicializada como:

```
canal_t chan121 = {
    0x1,                                // Bloqueado
    121,                                 // Número do canal: 121
    79,                                  // Preço: R$ 79
    "Rede Globo",                         // Nome da emissora
    "Vale a Pena Ver de Novo!"           // Nome do programa
};
```

0 – Sair

Esta opção deve encerrar o programa.

1 – Exibir programação (listas canais)

Esta operação percorre a lista de canais, imprimindo na tela as informações pertinentes a cada um:

Saída:

<Número> <Bloqueado> <Preço> <Nome> <Programa>

2 – Adicionar novo canal na lista

Esta operação permite o usuário digitar todos os dados necessários para seu programa montar uma estrutura *canal_t* e adicioná-la na lista.

Entrada:

<Número>
<Bloqueado>
<Preço>
<Nome>
<Programa>

3 – Remover canal da lista

Esta operação espera que o usuário informe o número do canal que deve ser removido da lista. Se o canal não existir, nada precisa ser feito.

Entrada:

<Número>

4 – Exibir apenas os canais bloqueados

Esta operação percorre a lista de canais, imprimindo na tela apenas os que estiverem marcados como bloqueados.

Saída:

<Número> <Bloqueado> <Preço> <Nome> <Programa>

5 – Procurar canal (número)

Esta operação pesquisa na lista de canais pelo número do canal informado pelo usuário. Caso ele existir, imprime seus dados na tela:

Entrada:

<Número>

Saída:

<Número> <Bloqueado> <Preço> <Nome> <Programa>

6 – Procurar canal (nome)

Esta operação pesquisa na lista de canais pelo nome da emissora do canal informado pelo usuário. Caso ele existir, imprime seus dados na tela:

Entrada:

<Nome>

Saída:

<Número> <Bloqueado> <Preço> <Nome> <Programa>

7 – Gravar lista no disco (ASCII)

Esta operação faz um *dump* (despeja) da lista de canais com seus respectivos dados em um *arquivo de texto* no disco rígido para serem recuperadas posteriormente através da operação 8.

8 – Carregar lista do disco (ASCII)

Esta operação lê um determinado *arquivo de texto* do disco rígido para montar a lista de canais da aplicação.

9 – Gravar lista no disco (binário)

Esta operação faz um *dump* (despeja) a lista de canais com seus respectivos dados em um *arquivo binário* no disco rígido para serem recuperadas posteriormente através da operação 10. Para realizar esta tarefa é essencial obedecer a quantidade de bits/bytes designada para cada membro da estrutura *canal_t*. Desta forma todas as informações de um canal serão agrupadas em um bloco de memória com 1156 bytes (9248 bits). O esquema abaixo ilustra o formato desse bloco de memória:

1 bit	15 bits	16 bits	1024 bits	8192 bits
1	2–16	17–32	33–1056	1057–9249
<i>bloq</i>	<i>num</i>	<i>preço</i>	<i>nome</i>	<i>programa</i>

Um arquivo binário pode conter vários blocos de 1156 bytes sequenciais, indicando que o arquivo contém uma lista de canais que possui mais que um canal armazenado. Serão disponibilizados no mural dois arquivos binários para ajudar você a [testar sua implementação de leitura/gravação](#).

10 – Carregar lista do disco (binário)

Esta operação lê um determinado *arquivo binário* do disco rígido para montar a lista de canais da aplicação.