Programação de Computadores

Aula #12

Bibliotecas Úteis em C

O que são bibliotecas mesmo? Quais são exemplos de bibliotecas úteis para a linguagem C?

Ciência da Computação - BCC e IBM - 2024/01
Prof. Vinícius Fülber Garcia

Bibliotecas

Em um contexto geral, uma biblioteca pode ser definida como...

"Coleção de livros, pública ou privada, classificados segundo algum critério, com o objetivo de conservá-los e de facilitar a consulta e o estudo" — **Michaelis**

No contexto específico da programação, podemos entender uma biblioteca como uma coleção de recursos e ferramentas (preferencialmente documentados) disponíveis para o uso conforme a necessidade do programador.

Bibliotecas em C

O uso de bibliotecas já é comum nos nossos programas! Além disso, já conseguimos criar nossas próprias “““““bibliotecas””””” via arquivos de código (.c) e cabeçalhos (.h).

Porém, o universo de bibliotecas de uma linguagem de programação é gigantesco, não se resumindo às bibliotecas nativas da linguagem e às pessoais.

Quase sempre, reinventar a roda não é um bom negócio!

Bibliotecas em C

Mas onde posso encontrar novas bibliotecas?

A maneira mais simples é procurar por repositórios online. Existem alguns projetos que visam agrupar bibliotecas, como, por exemplo:

- CCAN (https://ccodearchive.net/)
- GitHub (https://github.com/topics/c-library)
- Projetos independentes
- ...

LibC

A biblioteca padrão (e consequentemente nativa) da linguagem C, que traz diversas funções básicas e de comum utilização, é a chamada **LibC**.

Já estamos acostumados a utilizar muitos de seus módulos e funções!

- Interface com o sistema operacional
  - Alocação de memória
  - Acesso a arquivos
  - Acesso a *streams*
- Manipulação de caracteres e *strings*
- Ordenação e busca
- Operações matemáticas
- Conversões numéricas
- Geração de números aleatórios
- Operações com números complexos
- Manipulação de datas e horas
- ...

LibC

Ainda, é importante destacar a existência de uma extensão da LibC que trata funções específicas de sistema operacionais padrão POSIX:

POSIX C Library

E para Windows?

Subsystem for UNIX-based Applications (SUA)

Windows Subsystem for Linux (WSL)

- Operações de rede
- Tratamento de sinais e eventos do SO
- Operações de entrada/saída assíncronas
- Filas de mensagens
- Semáforos
- Threads
- ...

LibC

E a tal da GNU C *Library* (GLibC)?

Sistemas UNIX como o Linux e o FreeBSD usam geralmente a implementação da LibC construída pelo projeto GNU, chamada GNU C *Library*, ou simplesmente GLibC.

A GLibC implementa as funcionalidades da LibC padrão e da extensão POSIX, mas traz também um grande conjunto de extensões que não estão disponíveis em outras implementações.

GLib e SGLib

Existem bibliotecas, como a GLib e SGLib, focadas em prover estruturas genéricas de dados para serem usadas em C.

Os mais famosos exemplos dessas bibliotecas são a GLib e a SGLib!

Exemplos de estruturas de dados providas pela GLib e SGLib são:

- Listas ligadas
- Árvores
- Tabelas *hash*

GLib e SGLib

```c
#include <glib.h>

int main() {
   GList* frutas = NULL;

   frutas = g_list_append(frutas, "Maçã");
   frutas = g_list_append(frutas, "Banana");

   frutas = g_list_remove(frutas, "Maçã");

   g_list_free(frutas);

   return 0;
}
```

O código ao lado exemplifica a **criação e uso básico de uma lista** proveniente da biblioteca GLib.

Instalação:

sudo apt-get install libglib2.0-dev

Compilação:

gcc $(pkg-config --cflags glib-2.0) -o test teste.c $(pkg-config --libs glib-2.0)

SDL e Allegro

Outras bibliotecas facilitam a criação e **manipulação de recursos gráficos** a partir da linguagem C. Entre essas, destacam-se a *Simple DirectMedia Layer* (SDL) e a Allegro.

Essas bibliotecas são multiplataforma, sendo a SDL um tanto mais completa e complexa (opinião) que a Allegro!

SDL e Allegro

```c
#include <allegro5/allegro.h>

int main() {

   al_init();
   ALLEGRO_DISPLAY* display =
al_create_display(640, 480);
   al_clear_to_color(al_map_rgb(255, 255,
255));
   al_flip_display();
   al_rest(5.0);
   al_destroy_display(display);

   return 0;
}
```

O código ao lado exemplifica a **criação de uma janela simples** através da biblioteca Allegro.

Instalação:

sudo apt install liballegro5-dev

Compilação:

gcc $(pkg-config --cflags allegro-5) teste.c -o test $(pkg-config --libs allegro-5)

NCurses e GTK

A comunicação com o usuário é um ponto fundamental em qualquer programa! Devido a isso, existem bibliotecas direcionadas para a criação e gerenciamento de interfaces com o usuário.

Entre essas bibliotecas, colocamos em evidência a NCurses e a GIMP *ToolKit* (GTK). A NCurses é voltada para a criação de interfaces em terminal de texto, enquanto a GTK é voltada para a manipulação de janelas.

NCurses e GTK

```c
#include <gtk/gtk.h>

int main(int argc, char *argv[]) {
   GtkWidget *window;
   GtkWidget *image;

   gtk_init(&argc, &argv);
   window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
   g_signal_connect(window, "destroy",
G_CALLBACK(gtk_main_quit), NULL);
   image = gtk_image_new_from_file( "image.jpg");
   gtk_container_add(GTK_CONTAINER(window),
image);
   gtk_widget_show_all(window);
   gtk_main();

   return 0;
}
```

O código ao lado exemplifica a **criação de uma janela simples** com apresentação da imagem local chamada “image.jpg” pelo GTK.

Instalação:

sudo apt-get install libgtk-3-dev

Compilação:

gcc $(pkg-config --cflags gtk+-3.0) teste.c -o test $(pkg-config --libs gtk+-3.0)

GDBM e SQLite

Estruturas de armazenamento definem outro ponto de atenção durante a programação de um sistema, sejam essas estruturas definidas em memória principal ou secundária.

A biblioteca GNU DBM (GDBM) permite criar estruturas de armazenamento manipuladas em memória principal baseadas em relações de chave/valor; já a biblioteca SQLite, permite executar operações típicas de gerenciamento em bases de dados relacionais padrão SQL.

GDBM e SQLite

```c
#include <stdio.h>
#include <stdlib.h>
#include <gdbm.h>
int main(void) {
   GDBM_FILE db;
   datum key, value;
   db = gdbm_open( "meu_banco.db", 0, GDBM_WRCREAT, 0666,
NULL);
   key.dptr = "chave";
   key.dsize = sizeof("chave");
   value.dptr = "valor";
   value.dsize = sizeof("valor");
   gdbm_store(db, key, value, GDBM_REPLACE);
   key.dptr = "chave";
   key.dsize = sizeof("chave");
   value = gdbm_fetch(db, key);
   printf("Valor da chave 'chave': %s\n" , value.dptr);
   gdbm_close(db);
   return EXIT_SUCCESS;
}
```

O código ao lado exemplifica a **criação de uma base de dados GDBM**, a inserção de uma chave/valor e sua posterior exibição.

Instalação:

sudo apt-get install libgdbm6 libgdbm-dev

Compilação:

gcc teste.c -o test -lgdbm

GSL e OpenCV

Para aplicações científicas, também podemos contar com algumas bibliotecas, como a GNU *Scientific Library* (GSL) e a *Open Computer Vision* (OpenCV).

A GSL provê várias operações relacionadas a, por exemplo, números complexos e métodos numéricos; enquanto a OpenCV permite realizar processamento de imagens por meio de diversas funções diferentes.

GSL e OpenCV

```c
#include <stdio.h>
#include <gsl/gsl_poly.h>
int main(void) {
   double coefs[3] = {1.0, -3.0, 2.0};
   double raizes[2];
   int num_raizes;
   num_raizes = gsl_poly_solve_quadratic(coefs[0],
coefs[1], coefs[2], &raizes[0], &raizes[1]);

   if (num_raizes == 2) {
       printf("As raízes do polinômio são: %f e %f\n",
raizes[0], raizes[1]);
   } else if (num_raizes == 1) {
       printf("O polinômio possui uma única raiz real:
%f\n", raizes[0]);
   } else {
       printf("O polinômio não possui raízes reais\n");
   }
   return 0;
}
```

O código ao lado exemplifica a **obtenção das raízes de um polinômio de segundo grau** a partir da biblioteca GSL.

Instalação:

sudo apt-get install libgsl-dev

Compilação:

gcc $(pkg-config --cflags gsl) teste.c -o test $(pkg-config --libs gsl)

O Drama da Compilação

Às vezes, dependendo de como uma biblioteca é provida, os processos de compilação e ligação podem ficar “mais complexos”!

Bibliotecas Estáticas Vs. Bibliotecas Compartilhadas/Dinâmicas

Para bibliotecas compartilhadas, podemos facilitar um pouco as coisas utilizando, como vimos nos exemplos, o utilitário *pkg-config*!

--cflags **--libs**

Exercício #12

Escreva um programa em C que solicita ao usuário que digite dois números inteiros (para facilitar), que representam a parte real e imaginária de um número complexo, respectivamente. Em seguida, **utilize a biblioteca GSL para calcular a potenciação complexa** desse número (expoente 2) e exiba o resultado na tela. O programa deve realizar a validação de entrada do usuário, garantindo que apenas valores numéricos inteiros são aceitos.

Dicas:

gsl_complex_rect: função de definição de um número imaginário
gsl_complex_pow_real: função de potenciação de um número imaginário
gsl_complex: estrutura de dados para números imaginários

Obrigado!

Vinícius Fülber Garcia
inf.ufpr.br/vinicius
vinicius@inf.ufpr.br