

# Лабораторная работа № 7

## libcurl

### Задание 1

Постановка задачи:

Отправка HTTPзапроса типа GET с помощью libcurl. Напишите программу, которая будет отправлять GETзапрос на определённый URL с использованием библиотеки libcurl. Затем пусть программа выводит полученный HTMLкод или информацию о статусе ответа на экран. URL сайта подберите самостоятельно.

Математическая модель:

-

Список идентификаторов:


Имя	Тип	Назначение
curl	CURL *	Объект curl.
res	CURLcode	Возвращаемое curl значение.

Код программы:

```
1 #include <curl/curl.h>
2 #include <curl/easy.h>
3 #include <stdio.h>
4
5 int
6 main(void)
7 {
8     CURL *curl;
9     CURLcode res;
10
11     curl = curl_easy_init();
12     if(!curl)
13     {
14         fprintf(stderr, "Curl init failed!");
15         return 1;
16     }
17     curl_easy_setopt(curl, CURLOPT_URL , "https://wttr.in/?1R");
18
19     res = curl_easy_perform(curl);
20
21     if(res != CURLE_OK)
22         fprintf(stderr, "curl failed with %s\n", curl_easy_strerror(res));
23     else
24         fprintf(stdout, "curl succeded with code %d\n", res);
25
26     curl_easy_cleanup(curl);
27     return 0;
28 }
```





Результат выполненной работы:

Weather report: St Petersburg, Russia



Partly cloudy  
+13(12) °C  
↓ 15 km/h  
10 km  
0.0 mm

Sat 27 Apr

Morning	Noon	Evening	Night
<div><div>Sunny +9(8) °C → 7-8 km/h 10 km 0.0 mm   0%</div></div>	<div><div>Sunny +11(10) °C → 10-12 km/h 10 km 0.0 mm   0%</div></div>	<div><div>Sunny 10 °C □ 5-7 km/h 10 km 0.0 mm   0%</div></div>	<div><div>Partly Cloudy +7(5) °C ← 8-16 km/h 10 km 0.0 mm   0%</div></div>

Follow @igor\_chubin for wtr.in updates

curl succeeded with code 0

Задание 2

Постановка задачи:

Распечатка HTTP заголовков GET запросов с libcurl. Написать программу, которая отправляет HTTP GET запрос на определенный URL с использованием библиотеки libcurl. Затем программа извлекает и выводит на экран все заголовки HTTPответа, такие как Content-Type, Content-Length, Server или иные другие. Подберите URL сайта, чтобы распечатать как можно больше различных HTTP заголовков (HTTP headers).

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
curl	CURL *	Объект curl.
res	CURLcode	Возвращаемое curl значение.

Код программы:

```

1 #include <curl/curl.h>
2 #include <curl/easy.h>
3 #include <stdio.h>
4
5 size_t write_function(char *buffer, size_t size, size_t nitems, void *userdata)
6 ;
7 int
8 main(void)
9 {
10     CURL *curl;
11     CURLcode res;
12
13     curl = curl_easy_init();
14     if(!curl)
15     {
16         fprintf(stderr, "Curl init failed!");
17         return 1;
18     }
19     curl_easy_setopt(curl, CURLOPT_URL , "https://moodle.herzen.spb.ru/pluginfile.php/1671864/mod_resource/content/1/%D0%98%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%20-%20libcurl.pdf");
20     curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_function);
21     curl_easy_setopt(curl, CURLOPT_HEADERDATA, stdout);
22     curl_easy_setopt(curl, CURLOPT_WRITEDATA, NULL);
23     res = curl_easy_perform(curl);
24
25     if(res != CURLE_OK)
26         fprintf(stderr, "curl failed with %s\n", curl_easy_strerror(res));
27     else
28         fprintf(stdout, "curl succeeded with code %d\n", res);
29
30     curl_easy_cleanup(curl);
31     return 0;
32 }
33
34 size_t
35 write_function(char *buffer, size_t size, size_t nitems, void * stream)
36 {
37     if(stream != NULL)
38         fwrite(buffer, size, nitems, stream);
39     return size*nitems;
40 }

```

Результат выполненной работы:

```

HTTP/1.1 303 See Other
Server: nginx
Date: Sat, 27 Apr 2024 12:58:56 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Set-Cookie: MoodleSession=uqsa0fdd6no0sli4brn0d8nbji; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
X-Redirect-By: Moodle /lib/moodlelib.php:2787
Location: https://moodle.herzen.spb.ru/login/index.php
Content-Language: ru

curl succeeded with code 0

```

### Задание 3

Постановка задачи:

Задача на загрузку файлов с libcurl: Напишите программу, которая будет загружать файл с удаленного сервера с помощью libcurl. Пусть программа принимает URL для загрузки и путь для сохранения файла на локальном компьютере. Например это может быть картинка с соответствующим URL. После загрузки файла программа должна проверять статус загрузки и выводить сообщение об успешной или неуспешной загрузке.

Математическая модель:

-

Список идентификаторов:

Имя	Тип	Назначение
filepath	char *	Передаётся как аргумент программе, путь по которому будет сохранен файл.
url	char *	Передаётся как аргумент программе, ссылка на файл.
curl	CURL *	Объект curl.
res	CURLcode	Возвращаемое curl значение.
file	FILE *	Указатель на файл в который будет записана загруженная информация.
tmp	char *	Переменная для временного хранения указателя на строку разбитую по /. Нужно для того, чтобы получить дефолтное название файла из конца ссылки.

## Код программы:

```
1 #include <curl/curl.h>
2 #include <curl/easy.h>
3
4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <string.h>
7
8 void usage();
9 char *get_default_filepath(char *);
10
11
12 int
13 main(int argc, char** argv)
14 {
15     char *filepath = NULL;
16     char *url = NULL;
17
18     if(argc == 2){
19         url = argv[1];
20         filepath = get_default_filepath(url); //filepath is a pointer inside url
21     }else if(argc == 3){
22         url = argv[1];
23         filepath = argv[2];
24     }else
25         usage();
26
27     printf("url:%s\nfilepath:%s\n", url, filepath);
28
29     CURL *curl = curl_easy_init();
30     CURLcode res;
31
32     if(!curl)
33     {
34         fprintf(stderr, "Curl init failled, baling!");
35         exit(1);
36     }
37
38     FILE *file;
39     file = fopen(filepath, "wb");
40
41     curl_easy_setopt(curl, CURLOPT_URL, url);
42     curl_easy_setopt(curl, CURLOPT_WRITEDATA, file);
43
44
45     if(!file){
46         fprintf(stderr, "Could not open file!");
47         exit(1);
48     }
49
50     res = curl_easy_perform(curl);
51
```

```

49     res = curl_easy_perform(curl);
50
51
52     if(res != CURLE_OK)
53         fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
54     else
55         fprintf(stdout, "curl returned %s; Downloaded %s \n", curl_easy_strerror(res), filepath);
56
57     fclose(file);
58     curl_easy_cleanup(curl);
59     return 0;
60 }
61
62
63 char* get_default_filepath(char *url) {
64     char *filepath = strdup(url);
65     char *tmp = strtok(filepath, "/");
66     while(tmp != NULL) {
67         filepath = tmp;
68         tmp = strtok(NULL, "/");
69     }
70     return filepath;
71 }
72
73 void
74 usage(){
75     puts("usage: <url> <file path>(optional, default is ../<filename on url>)");
76     exit(0);
77 }

```

## Результат работы:

```

>./a.out https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.7.tar.xz ../new_kernel.tar.gz
url:https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.7.tar.xz
filepath:../new_kernel.tar.gz
curl returned No error; Downloaded ../new_kernel.tar.gz
>du -h ../new_kernel.tar.gz
136M    ../new_kernel.tar.gz

```