**Язык программирования C++**

*Отчет*

|  |  |
| --- | --- |
| Лабораторная работа 1 | *17.09.2019* |

Основы программирования для Linux

*Задача №1*

**1. Условия задачи**

*Раздел №5.7.1 из курса “Основы программирования для Linux”.* Разработайте утилиту командной строки, принимающую на вход имя хоста и печатающую в стандартный поток вывода список IP-адресов, ассоциированных с данным именем.

**2. Алгоритм. Блок-схема**

**3. Исходный код программы**

#include <stdio.h>

#include <stdlib.h>

#include <netdb.h>

#include <string.h>

#include <errno.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

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

{

if (argc != 2)

return -1;

int i;

struct hostent \*h;

struct in\_addr \*\*addr\_list;

h = gethostbyname(argv[1]);

if (!h) {

fprintf(stderr, "gethostbyname: %s\n", strerror(errno));

exit(1);

}

addr\_list = (struct in\_addr \*\*)h->h\_addr\_list;

for(i = 0; addr\_list[i] != NULL; i++) {

printf("%s\n", inet\_ntoa(\*addr\_list[i]));

}

return 0;

}

**4. Формат входных и выходных данных**

Формат входных данных

Программа принимает на входе строку.

Формат выходных данных

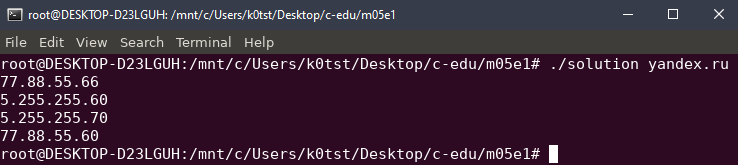
Программа выводит в стандартный поток вывода IP-адреса, причем каждый адрес должен быть выведен на отдельной строчке, оканчивающейся символом конца строки.

**5. Результат работы программы**



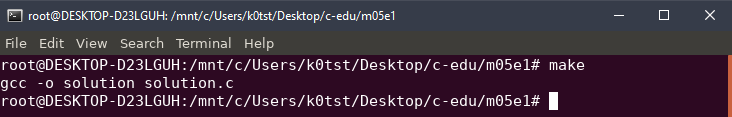
…





**6. Выводы и комментарии к решению задачи**

* Для работы с сокетами и пакетами используются библиотеки: netdb.h, socket.h, in.h, inet.h.



*Задача №2*

**1. Условия задачи**

*Раздел №5.7.2 из курса “Основы программирования для Linux”.* Разработайте UDP сервер для логирования (сбора и записи) сообщений. Сервер при запуске принимает параметр -- номер порта, на котором будут приниматься запросы. Каждый запрос приходящий к серверу - строка, не превышающая размер 5K. Задача сервера -- выводить приходящие строки в стандартный поток ввода-вывода, завершающиеся символом конца строки.

Сервер завершает работу, когда получает строку содержащую только текст 'OFF', и заканчивающуюся символом перевода строки '\n'.

Сервер использует локальный адрес 127.0.0.1.

**2. Алгоритм. Блок-схема**

**3. Исходный код программы**

#include <stdio.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/ip.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <string.h>

#define BUFLEN 5120

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

{

if (argc != 2) {

fprintf(stderr, "Usage: %s <port>\n", argv[0]);

exit(1);

}

int sock, port;

struct sockaddr\_in local;

struct sockaddr remote;

socklen\_t slen = sizeof(remote);

char buf[BUFLEN];

port = atoi(argv[1]);

sock = socket(AF\_INET, SOCK\_DGRAM, 0);

inet\_aton("127.0.0.1", &local.sin\_addr);

local.sin\_port = htons(port);

local.sin\_family = AF\_INET;

bind(sock, (struct sockaddr \*) &local, sizeof(local));

while (1) {

memset(buf, 0, BUFLEN);

recvfrom(sock, buf, BUFLEN, 0, &remote, &slen);

if (strncmp("OFF\n", buf, BUFLEN) == 0)

break;

else

buf[BUFLEN - 1] = '\0';

printf("%s\n", buf);

}

return 0;

}

**4. Формат входных и выходных данных**

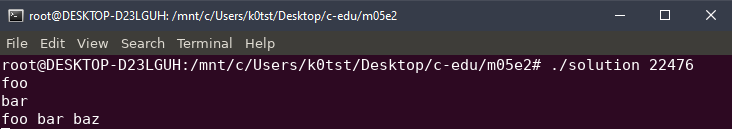
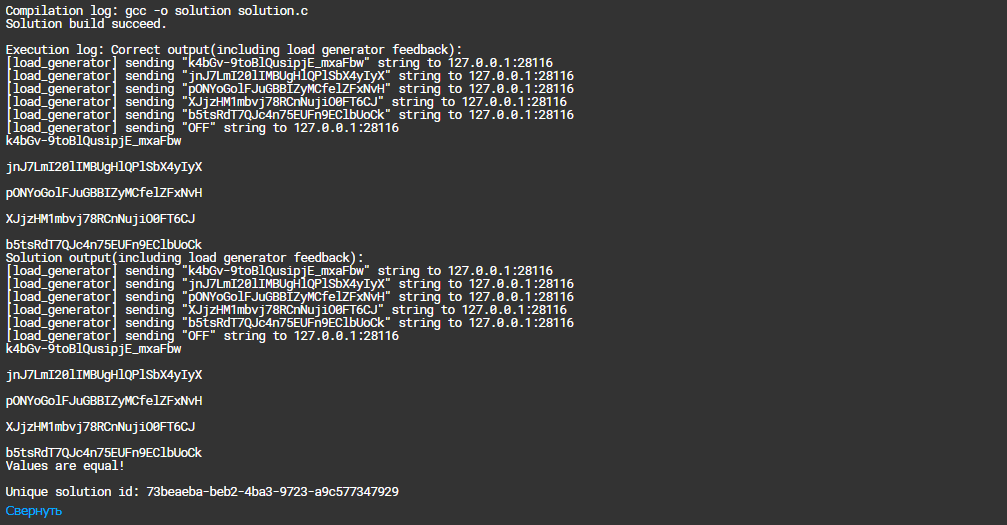
Формат входных данных

Передача аргумента 8 в функцию.

Формат выходных данных

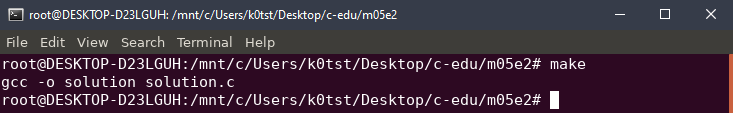
Программа выводит в стандартный поток строки, причем каждая строка должна оканчиваться символом конца строки.

**5. Результат работы программы**



**6. Выводы и комментарии к решению задачи**

* Программа не вернула и не приняла передаваемые ей данные.



*Задача №3*

**1. Условия задачи**

*Раздел №5.7.3 из курса “Основы программирования для Linux”.* Разработайте TCP сервер, предназначенный для сортировки символов в строке. Сервер получает на вход строки, оканчивающиеся символом '\0', сортирует в них символы в порядке убывания ASCII-кодов и отсылает обратно на клиент, в виде строки заканчивающейся '\0'. Завершение работы сервера происходит после получения строки, содержащей только 'OFF'.

При старте сервер получает на вход номер порта в качестве параметра командной строки. bind производится на адресе 127.0.0.1.

**2. Алгоритм. Блок-схема**

**3. Исходный код программы**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>

#include <string.h>

#include <strings.h>

#include <unistd.h>

#define MAXLINE 1000

int cmp\_func(const void \* a, const void \* b)

{

return \*(char \*)b - \*(char \*)a;

}

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

{

if (argc != 2) {

fprintf(stderr, "Usage: %s <port>\n", argv[0]);

exit(1);

}

char buf[BUFSIZ];

int listen\_fd, comm\_fd;

int port;

struct sockaddr\_in servaddr;

listen\_fd = socket(AF\_INET, SOCK\_STREAM, 0);

bzero( &servaddr, sizeof(servaddr) );

port = atoi(argv[1]);

inet\_aton("127.0.0.1", &servaddr.sin\_addr);

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(port);

bind(listen\_fd, (struct sockaddr \*) &servaddr, sizeof(servaddr));

listen(listen\_fd, 10);

comm\_fd = accept(listen\_fd, (struct sockaddr\*) NULL, NULL);

while (1) {

bzero(buf, BUFSIZ);

read(comm\_fd, buf, BUFSIZ);

if (strncmp("OFF", buf, 3) == 0)

break;

qsort(buf, (size\_t) strlen(buf), (size\_t) sizeof(char), cmp\_func);

write(comm\_fd, buf, strlen(buf)+1);

}

return 0;

}

**4. Формат входных и выходных данных**

Формат входных данных

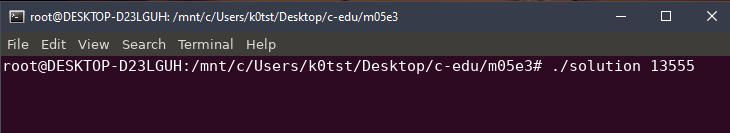
Передача аргумента в программу.

Формат выходных данных

Программа выводит в стандартный поток строки, причем каждая строка должна оканчиваться символом конца строки.

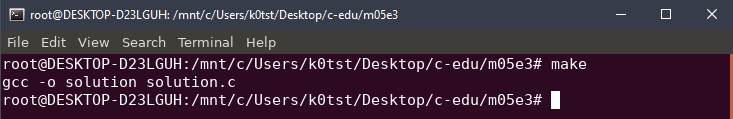
**5. Результат работы программы**





**6. Выводы и комментарии к решению задачи**

* Для корректной работы программы необходимо использовать Docker.



*Выводы по лабораторной работе 1*

**1.** Решены 3 задач по теме «*Основы программирования для Linux*»:

1) Host resolver,

2) UDP log service,

3) TCP string sorter.

СПИСОК ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Основы программирования для Linux. – Режим доступа: <https://stepik.org/lesson/26302/step/1?unit=8180>, свободный. Загл. с экрана. – Дата обращения: 10.09.2019