上海工程技术大学

实 验 报 告

实验六　Linux 下的 C 语言编程

1. 实验要求

（1）熟悉 Linux 环境下 C 语言应用程序开发的基本过程；

（2）熟悉基本库函数的使用；

（3）具有初步的应用程序设计能力。

2. 实验内容

1） 有三个程序 print.h、hello.c 和 print.c，它们的代码在下面已被给出：

① 创建一个目录文件 ctest，把程序 print.h、hello.c 和 print.c 都放在 ctest 中；然后，编译、执行这三个程序；

② 在 ctest 中创建一个目录文件 ctest2，并把程序 print.h 移到 ctest2 中；然后，编译、执行这三个程序。

源程序 print.h：

```
#define borderchar '*'
void my_print(char *);
```

源程序 hello.c：

```
#include <print.h>
main()
{
 char my_string[]="Hello world!";
 my_print(my_string);
}
```

源程序 print.c：

```
#include<stdio.h>
#include<string.h>
#include "print.h"
void my_print(char * str)
{
int i;
```

```
for (i=0;i<strlen(str)+4;i++)
   printf("%c",borderchar);
printf("\n");
printf("%c %s %c\n",borderchar,str,borderchar);
for (i=0;i<strlen(str)+4;i++)
   printf("%c",borderchar);
printf("\n");
   }
```

2） 下面的程序 greeting.c 实现的功能是：先显示字符串“hello there”，再将该字符串反序输出。请编译、执行程序 greeting.c，检查它的输出结果。如果输出结果不符合要求，请通过调试找出导致问题的语句，并进行修正：

程序 greeting.c 如下：

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
   void my_print(char * string);
   void my_print2(char * string);
   char my_string[]="hello there";
   my_print(my_string);
   my_print2(my_string);
   return 0;
}

void my_print(char * string)
{
   printf("The string is %s\n",string);
}

void my_print2(char * string)
{
   char *string2;
   int size,i;
   size=strlen(string);
   string2=(char *)malloc(size+1);

   for (i=0;i<size;i++)
string2[size-i]=string[i];

   string2[size+1]='\0';
   printf("The string printed backward is %s\n",string2);
```

}

3) 用 C 程序设计和实现某一磁盘调度算法;

4) 思考题（选做）

编译、运行下面的使用 TCP 协议实现的 client/server 网络通信程序。

TCP 服务器的参考代码 tcp_server.c:

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>          //inet_ntoa()函数的头文件
#define portnumber 3333     //定义端口号：（0-1024 为保留端口号，最好不要用）

int main(int argc, char *argv[])
{
   int sockfd,new_fd;
   struct sockaddr_in server_addr; //描述服务器地址
   struct sockaddr_in client_addr; //描述客户端地址
   int sin_size;
   char hello[]="Hello! Are You Fine?\n";


    /* 服务器端开始建立 sockfd 描述符 */
   if((sockfd=socket(AF_INET,SOCK_STREAM,0))==-1)

//AF_INET:IPV4;SOCK_STREAM:TCP
   {
       fprintf(stderr,"Socket error:%s\n\a",strerror(errno));
       exit(1);
    }

    /* 服务器端填充 sockaddr 结构 */
    bzero(&server_addr,sizeof(struct sockaddr_in));     // 初始化,置 0
    server_addr.sin_family=AF_INET; // Internet
    server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
      // (将本机器上的 long 数据转化为网络上的 long 数据)和任何主机通信
      //INADDR_ANY 表示可以接收任意 IP 地址的数据，即绑定到所有的 IP
      //server_addr.sin_addr.s_addr=inet_addr("192.168.1.1");
      //用于绑定到一个固定 IP,inet_addr 用于把数字加格式的 ip 转化为整形 ip
    server_addr.sin_port=htons(portnumber);
```

```
        // (将本机器上的 short 数据转化为网络上的 short 数据)端口号

  /*  捆绑 sockfd 描述符到 IP 地址  */
  if(bind(sockfd,(struct sockaddr *)(&server_addr),sizeof(struct sockaddr))==-1)
        {
     fprintf(stderr,"Bind error:%s\n\a",strerror(errno));
           exit(1);
         }

  /*  设置允许连接的最大客户端数  */
        if(listen(sockfd,5)==-1)
        {
           fprintf(stderr,"Listen error:%s\n\a",strerror(errno));
     exit(1);
        }

while(1)
{
    /*  服务器阻塞,直到客户程序建立连接  */
          sin_size=sizeof(struct sockaddr_in);
          if((new_fd=accept(sockfd,(struct sockaddr *)(&client_addr),&sin_size))==-1)
          {
             fprintf(stderr,"Accept error:%s\n\a",strerror(errno));
             exit(1);
          }
           fprintf(stderr,"Server get connection
from %s\n",inet_ntoa(client_addr.sin_addr));
                  // 将网络地址转换成.字符串，并打印到输出终端
            //向客户端程序写入 hello 数组里的字符
           if(write(new_fd,hello,strlen(hello))==-1)
           {
         fprintf(stderr,"Write Error:%s\n",strerror(errno));
              exit(1);
           }
      /*  这个通讯已经结束  */
             close(new_fd);
             /*  循环下一个  */
        }

              /*  结束通讯  */
        close(sockfd);
              exit(0);
}
```

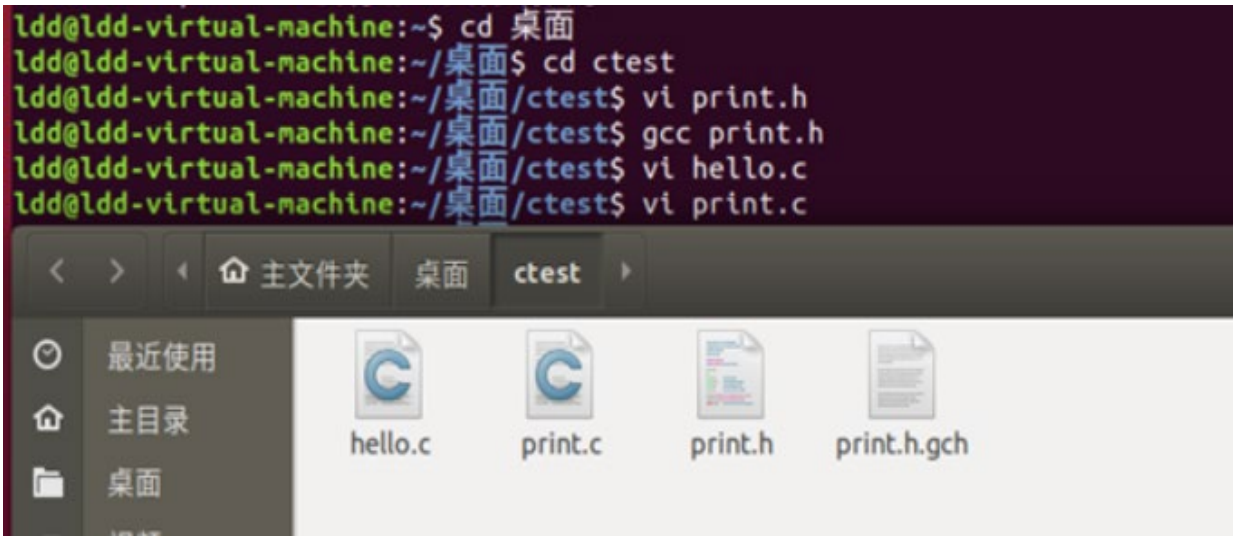
TCP 客户端的参考代码 tcp_client.c

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <arpa/inet.h>          //inet_ntoa()函数的头文件

#define portnumber 3333       //定义端口号：（0-1024 为保留端口号，最好不要用）

int main(int argc, char *argv[])
{
      int sockfd;
      char buffer[1024];
      struct sockaddr_in server_addr;       //描述服务器的地址
      struct hostent *host;
      int nbytes;

    /*  使用 hostname 查询 host  名字  */
      if(argc!=2)
      {
            fprintf(stderr,"Usage:%s hostname \a\n",argv[0]);
            exit(1);
      }

      if((host=gethostbyname(argv[1]))==NULL)
      {
            fprintf(stderr,"Gethostname error\n");
            exit(1);
      }

/*  客户程序开始建立  sockfd 描述符  */
if((sockfd=socket(AF_INET,SOCK_STREAM,0))==-1)
                                          // AF_INET:Internet;SOCK_STREAM:TCP
{
            fprintf(stderr,"Socket Error:%s\a\n",strerror(errno));
            exit(1);
        }

 /*  客户程序填充服务端的资料  */
        bzero(&server_addr,sizeof(server_addr));       //  初始化,置 0
```

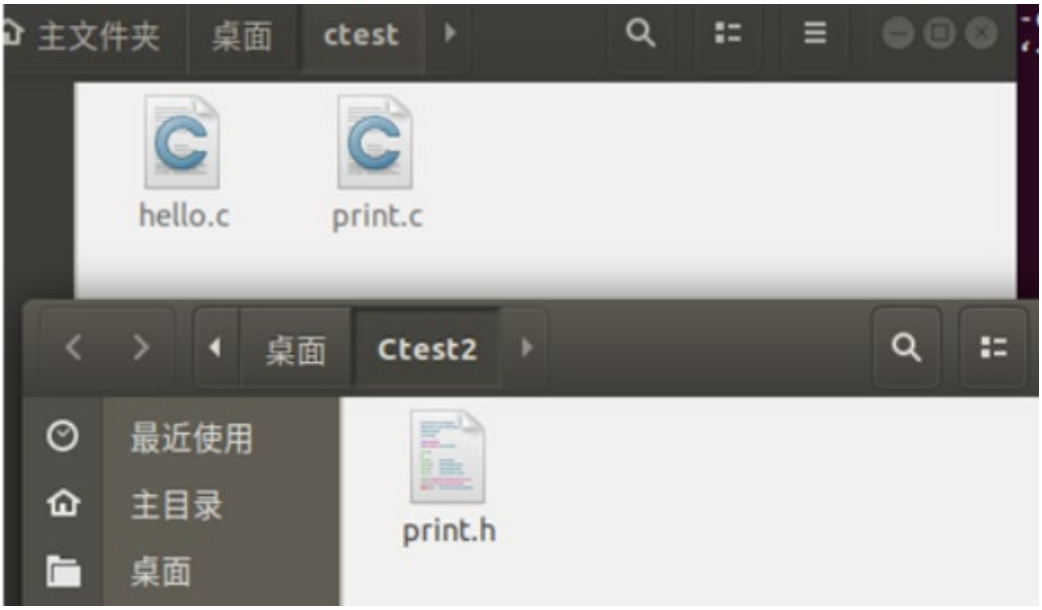
```
server_addr.sin_family=AF_INET; // IPV4
        server_addr.sin_port=htons(portnumber);
                        // (将本机器上的 short 数据转化为网络上的 short 数据)端口
号
      server_addr.sin_addr=*((struct in_addr *)host->h_addr);       // IP 地址

       /* 客户程序发起连接请求 */
if(connect(sockfd,(struct sockaddr *)(&server_addr),sizeof(struct sockaddr))==-1)
       {
          fprintf(stderr,"Connect Error:%s\a\n",strerror(errno));
          exit(1);
  }

  /* 连接成功了 */
 if((nbytes=read(sockfd,buffer,1024))==-1)
 {
          fprintf(stderr,"Read Error:%s\n",strerror(errno));
          exit(1);
  }
  buffer[nbytes]='\0';
  printf("I have received:%s\n",buffer);

 /* 结束通讯 */
 close(sockfd);
 exit(0);
}
```

3. 实验结果

（1）



```
ldd@ldd-virtual-machine:~/桌面/ctest$ vi hello.c
ldd@ldd-virtual-machine:~/桌面/ctest$ gcc *.c -o test
hello.c:2:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 main()
 ^~~~
ldd@ldd-virtual-machine:~/桌面/ctest$ ls
hello.c  print.c  print.h  test
ldd@ldd-virtual-machine:~/桌面/ctest$ ./test
****************
* Hello world! *
****************
ldd@ldd-virtual-machine:~/桌面/ctest$
```



文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```c
#include<stdio.h>
#include<string.h>
#include "/home/ldd/桌面/Ctest2/print.h"
void my_print(char * str)
{
int i;
for (i=0;i<strlen(str)+4;i++)
  printf("%c",borderchar);
printf("\n");
printf("%c %s %c\n",borderchar,str,borderchar);
for (i=0;i<strlen(str)+4;i++)
  printf("%c",borderchar);
printf("\n");
 }
```

```
ldd@ldd-virtual-machine:~/桌面/ctest$ vi print.c
ldd@ldd-virtual-machine:~/桌面/ctest$ gcc *c -o test
hello.c:2:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 main()
 ^~~~
ldd@ldd-virtual-machine:~/桌面/ctest$ ./test
*****************
* Hello world! *
*****************
```

（2）修改后代码：

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
	void my_print(char* string);
	void my_print2(char* string);
	char my_string[] = "hello there";
	my_print(my_string);
	my_print2(my_string);
	return 0;
}

void my_print(char* string)
{
	printf("The string is % s\n", string);
}

void my_print2(char* string)
{
	char* string2;
	int size, i;
	size = strlen(string);
	string2 = (char*)malloc(size + 1);

	for (i = 0; i < size; i++)
		string2[size-i-1] = string[i];

	string2[size] = '\0';
	printf("The string printed backward is % s\n", string2);
}
```

```
The string is hello there
The string printed backward is ereht olleh
```

（3）

代码：#include<stdio.h>

```
int num, sum, kai, max;
int m = 0;
int n = 0;
int s[100];
int s1[100];
int c1[50];
int c2[50];

void creat()
{
    printf("-------------------------------------------------\n");
    printf("请输入从哪个磁道开始：\t\n");
    printf("-------------------------------------------------\n");
    scanf("%d", &kai);
    printf("-------------------------------------------------\n");
    printf("请输入最长磁道号：\n");
    printf("-------------------------------------------------\n");
    scanf("%d", &max);
    printf("-------------------------------------------------\n");
    printf("请输入磁道的个数：\n");
    printf("-------------------------------------------------\n");
    scanf("%d", &num);
    for (int j = 0; j < num; j++)
```

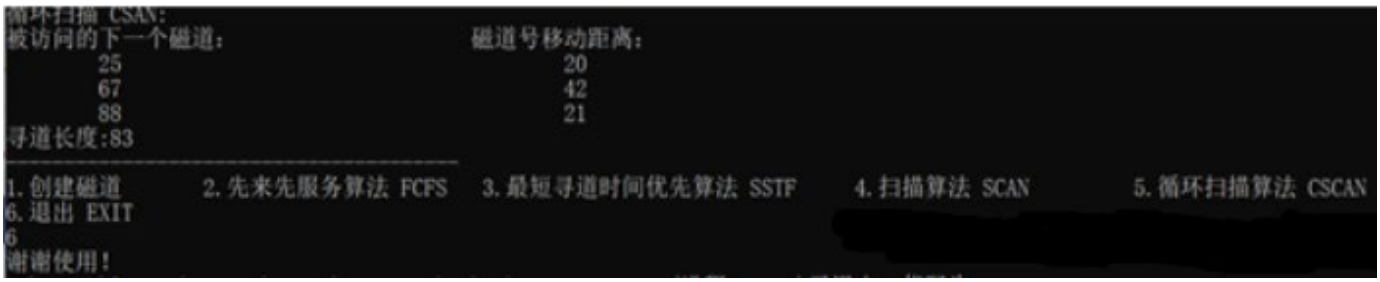
```
    {
        printf("请输入第%d 个磁道\n", j + 1);
        scanf("%d", &s[j]);
        if (s[j] > max)
        {
            printf("ERROR\n");
            break;
        }
        for (int i = 0; i < j; i++)
            if (s[j] == s[i])
                j--;
    }


    printf("被访问的下一个磁道\n");
    for (int i = 0; i < num; i++)
    {
        printf("\t%d\t\n", s[i]);
    }

    int su = kai;
    int t;
    for (int i = 0; i < num; i++)
        if (su > s[i])
            c1[m++] = s[i];
        else
            c2[n++] = s[i];
    for (int i = 0; i < m; i++)
        for (int j = i; j < m; j++)
            if (c1[i] < c1[j])
            {
                t = c1[i]; c1[i] = c1[j]; c1[j] = t;
            }
    for (int i = 0; i < n; i++)
        for (int j = i; j < n; j++)
            if (c2[i] > c2[j])
            {
                t = c2[i]; c2[i] = c2[j]; c2[j] = t;
            }

}

void FCFS()
{
```

```
    printf("先来先服务算法  FCFS\n");
    printf("被访问的下一个磁道\t\t\t 磁道号移动距离\n");
    int su = kai;
    sum = 0;
    for (int i = 0; i < num; i++)
    {
        if (su < s[i])
            s1[i] = s[i] - su;
        else
            s1[i] = su - s[i];
        su = s[i];
        sum += s1[i];
    }

    for (int i = 0; i < num; i++)
    {
        printf("\t%d\t\t\t\t\t%d\t\t\n", s[i], s1[i]);
    }
    printf("寻道长度：%d\n", sum);

}
void SSTF()
{
    printf("最短寻道时间优先算法  SSTF:\n");
    printf("被访问的下一个磁道\t\t\t 磁道号移动距离\n");
    int su = kai;
    int s2[100];
    sum = 0;

    for (int i = 0; i < m; i++)
        s2[i] = c1[i];
    for (int i = 0; i < n; i++)
        s2[i + m] = c2[i];
    for (int i = 0; i < num; i++)
    {
        if (su < s2[i])
            s1[i] = s2[i] - su;
        else
            s1[i] = su - s2[i];
        su = s2[i];
        sum += s1[i];
    }
    for (int i = 0; i < num; i++)
    {
```

```
            printf("\t%d\t\t\t\t\t%d\t\t\n", s2[i], s1[i]);
        }
        printf("寻道长度:%d\n", sum);
}
void SCAN()
{
        printf("扫描算法  SCAN:\n");
        printf("被访问的下一个磁道：\t\t\t 磁道号移动距离：\n");
        int su = kai;
        int s2[100];
        sum = 0;
        for (int i = 0; i < n; i++)
            s2[i] = c2[i];
        for (int i = 0; i < m; i++)
            s2[i + n] = c1[i];
        for (int i = 0; i < num; i++)
        {
            if (su < s2[i])
                s1[i] = s2[i] - su;
            else
                s1[i] = su - s2[i];
            su = s2[i];
            sum += s1[i];
        }
        for (int i = 0; i < num; i++)
        {
            printf("\t%d\t\t\t\t\t%d\t\t\n", s2[i], s1[i]);
        }
        printf("寻道长度：%d\n", sum);
}
void CSAN()
{
        printf("循环扫描  CSAN:\n");
        printf("被访问的下一个磁道：\t\t\t 磁道号移动距离：\n");
        int su = kai;
        int j = 0;
        int s2[100];
        sum = 0;
        for (int i = 0; i < n; i++)
            s2[i] = c2[i];
        for (int i = m - 1; i >= 0; j++, i--)
            s2[j + n] = c1[i];
        for (int i = 0; i < num; i++)
        {
```

```
            if (su < s2[i])
                s1[i] = s2[i] - su;
            else
                s1[i] = su - s2[i];
            su = s2[i];
            sum += s1[i];
        }
        for (int i = 0; i < num; i++)
        {
            printf("\t%d\t\t\t\t\t%d\t\t\n", s2[i], s1[i]);
        }
        printf("寻道长度:%d\n", sum);
}


void MENU()
{
        printf("磁盘调度\n");
        printf("---------------------------------------\n");
        printf("1.创建磁道\t 2.先来先服务 FCFS\t 3.最短寻道 SSTF\t 4.扫描算法 SCAN\t 5.循环扫描算法 CSCAN\t 6.退出 EXIT\n");
        int menuchoice;

        scanf("%d", &menuchoice);
        if (menuchoice != 1 && menuchoice != 6)
        {
            printf("请先创建磁道\n");
        }
        if (menuchoice == 6)
        {
            printf("谢谢使用！ ");
        }
        else
        {
            creat();
            printf("磁盘调度\n");
        P:printf("---------------------------------------\n");
            printf("1.创建磁道\t 2.先来先服务算法 FCFS\t 3.最短寻道时间优先算法 SSTF\t 4.扫描算法 SCAN\t 5.循环扫描算法 CSCAN\t 6.退出 EXIT\n");
            scanf("%d", &menuchoice);
            if (menuchoice > 6 || menuchoice < 1) {
                printf("未查询到操作指令！ \n");
                goto P;
            }
```

```
            switch (menuchoice)
            {
            case 2:
                FCFS();
                goto P;

            case 3:

                SSTF();
                goto P;
            case 4:

                SCAN();
                goto P;
            case 5:

                CSAN();
                goto P;

            case 6:

                printf("谢谢使用！ ");
                break;
            }
        }
}
int main()
{
    MENU();
    return 0;
}
```



4. 实验小结

通过本次实验，我熟悉了 Linux 环境下 C 语言应用程序开发的基本过程，熟悉了基本库函数的使用，感觉自己具有初步的应用程序设计能力。