

Experiment3 C programming experiment

Experimental purpose:

Further use the basic syntax of C programming language in Linux system, deepen the understanding of the knowledge.

(1) Task 1

Write a C program that uses standard I/O libraries to display the contents of text files. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile file.

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    char buf[1024] = { 0 };
    FILE* fp = fopen(argv[1], "r");
    if (argc < 2)
    {
        printf("please input source file!\n");
    }
    if (fp == NULL)
    {
        printf("open source %s failed\n", argv[1]);
        return -1;
    }
    while (fgets(buf, 1024, fp))
    {
        printf("%s\n", buf);
    }
    return 0;
}
```

Make sure your filename is c1.c

We can use the following makefile.

```
hello1:c1.o
gcc -o hello1 c1.o
c1.o:c1.c
gcc -c c1.c
clean:
rm -rf *.o
```

```
xuan@xuan-None: ~/B19040710
make: *** [makefile:2: hello1] 错误 1
xuan@xuan-None:~/B19040710$ make
gcc -o hello1 demo1.o
xuan@xuan-None:~/B19040710$ ls
王文轩2.txt  demo1.c  demo1.sh  demo3.sh  demo5.sh  makefile
B19040710.txt  demo1.o  demo2.sh  demo4.sh  hello1
xuan@xuan-None:~/B19040710$ ./hello1 王文轩2.txt
hello world

xuan@xuan-None:~/B19040710$ ./hello1 demo1.c
#include<stdio.h>

int main(int argc,char* argv[])
{
    char buf[1024]={0};

    FILE* fp=fopen(argv[1],"r");

    if(argc<2)
    {
```

(2) Task 2

Write a C program that displays all the file names in the current directory. The program is compiled and linked by the make tool, which requires the generation of the.o file first, and then the generation of the executable file, and the function of deleting the intermediate file (.o) in the makefile file.

include <stdio.h>

include <dirent.h>

include <sys/types.h>

```
int main(int argc, char* argv[])
{
    DIR* dirp;
    struct dirent* direntp;
    if ((dirp = opendir(argv[1])) == NULL) {
        printf("error\n");
        // exit(1);
    }
    while ((direntp = readdir(dirp)) != NULL)
        printf("%s\n", direntp->d_name);
    closedir(dirp);
    // exit(0);
}
```

Make sure your filename is c2.c

We can use the following makefile.

```
hello2:c2.o
```

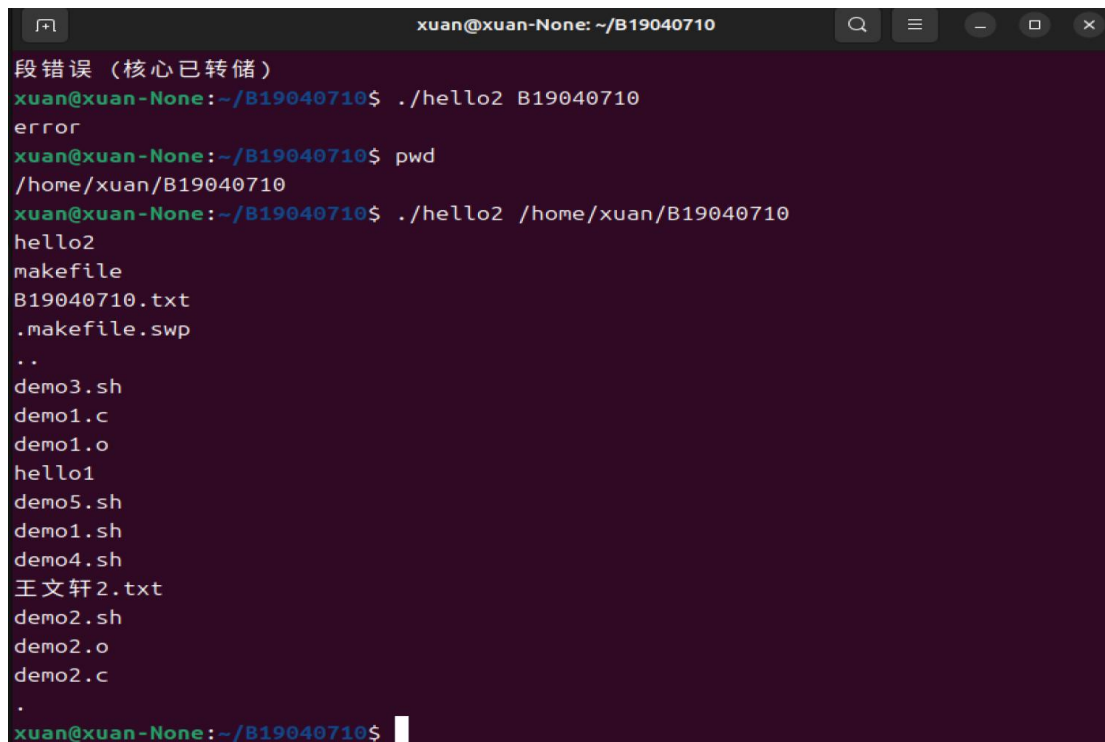
```
gcc -o hello1 c2.o
```

```
c2.o:c2.c
```

```
gcc -c c2.c
```

```
clean:
```

```
rm -rf *.o
```

A terminal window titled 'xuan@xuan-None: ~/B19040710' with standard window controls. The terminal shows a directory listing of the current directory, including files like 'demo3.sh', 'demo1.c', 'demo1.o', 'hello1', 'demo5.sh', 'demo1.sh', 'demo4.sh', '王文轩2.txt', 'demo2.sh', 'demo2.o', 'demo2.c', and a hidden file '.'. The prompt is 'xuan@xuan-None: ~/B19040710\$'. Above the prompt, there is a message in Chinese: '段错误 (核心已转储)' (Segmentation fault (core dumped)). Below the prompt, the command './hello2 B19040710' has been entered, followed by 'error' on the next line.

```
xuan@xuan-None: ~/B19040710$ ./hello2 B19040710
error
```

(3) Task 3

(3) Write a C program that changes the working directory of the current process. The program is compiled

and linked by the make tool, which requires the generation of the.o file first, and then the generation of the

executable file, and the function of deleting the intermediate file (.o) in the makefile file.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
int main(){
```

```
char buf[1024] = {0}; char buf2[1024]={0};
```

```
getcwd(buf, 1024);
```

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

```
if(chdir("/home")<0){
```

```
printf("error\n"); #
```

```
}
```

```
else
```

```

{
printf("success\n");
}
getcwd(buf2,1024);
printf("%s\n",buf2);
return 0;
}

```

Make sure your filename is c3.c

We can use the following makefile.

```
hello3:c3.o
```

```
gcc -o hello1 c3.o
```

```
c3.o:c3.c
```

```
gcc -c c3.c
```

```
clean:
```

```
rm -rf *.o
```

```

xuan@xuan-None: ~/B19040710
demo4.sh
王文轩2.txt
demo2.sh
demo2.o
demo2.c
.
xuan@xuan-None:~/B19040710$ touch demo3.c
xuan@xuan-None:~/B19040710$ open demo3.c
xuan@xuan-None:~/B19040710$ open makefile
xuan@xuan-None:~/B19040710$ ./hello3
bash: ./hello3: 没有那个文件或目录
xuan@xuan-None:~/B19040710$ open makefile
xuan@xuan-None:~/B19040710$ l
王文轩2.txt    demo1.o    demo2.o    demo3.sh*   hello1*
B19040710.txt  demo1.sh*  demo2.sh*  demo4.sh*   hello2*
demo1.c        demo2.c    demo3.c    demo5.sh*   makefile
xuan@xuan-None:~/B19040710$ open makefile
xuan@xuan-None:~/B19040710$ make
gcc -c demo3.c
gcc -o hello3 demo3.o
xuan@xuan-None:~/B19040710$ ./hello3
/home/xuan/B19040710
success/home
xuan@xuan-None:~/B19040710$

```