

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

(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 cl.c
We can use the following makefile.
hello1:cl.o
gcc -o hello1 cl.o
cl.o:cl.c

```

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

zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo B23040920
B23040920
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ touch c1.c c2.c c3.c
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ touch Makefile1 Makefile2 Makefile3
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > c1.c << 'EOF'
#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");
        return -1;
    }
    if (fp == NULL)
    {
        printf("open source %s failed\n", argv[1]);
        return -1;
    }
    while (fgets(buf,1024, fp))
    {
        printf("%s", buf);
    }
    return 0;
}
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > Makefile1 << 'EOF'
hello1: c1.o
    gcc -o hello1 c1.o

c1.o: c1.c
    gcc -c c1.c

clean:
    rm -rf *.o
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ make -f Makefile1
gcc -c c1.c
gcc -o hello1 c1.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ls -l hello1 c1.o
-rw-rw-r-- 1 zouxin zouxin 2136 12月 28 23:00 c1.o
-rwxrwxr-x 1 zouxin zouxin 16136 12月 28 23:00 hello1
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo "Hello, this is a test file." > test.txt
echo "This is the second line." >> test.txt
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./hello1 test.txt
Hello, this is a test file.
This is the second line.
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ make -f Makefile1 clean
rm -rf *.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo "First line" > file1.txt
echo "Second line" >> file1.txt
echo "Third line" >> file1.txt
./hello1 file1.txt
First line
Second line
Third line

```

(2) Task 2

- (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
```

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo B23040920
B23040920
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > c2.c << 'EOF'
#include <stdio.h>
#include <dirent.h>
#include <sys/types.h>

int main(int argc, char* argv[])
{
    DIR* dirp;
    struct dirent* direntp;

    if (argc < 2)
    {
        printf("Usage: %s <directory>\n", argv[0]);
        return 1;
    }

    if ((dirp = opendir(argv[1])) == NULL) {
        printf("error: cannot open directory %s\n", argv[1]);
        return 1;
    }

    while ((direntp = readdir(dirp)) != NULL)
        printf("%s\n", direntp->d_name);

    closedir(dirp);
    return 0;
}
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > Makefile2 << 'EOF'
hello2: c2.o
    gcc -o hello2 c2.o

c2.o: c2.c
    gcc -c c2.c

clean:
    rm -rf *.o
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ make -f Makefile2
gcc -c c2.c
gcc -o hello2 c2.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ls -l hello2 c2.o
-rw-rw-r-- 1 zouxin zouxin 1968 12月 28 23:04 c2.o
-rwxrwxr-x 1 zouxin zouxin 16136 12月 28 23:04 hello2
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./hello2 .
file1.txt
Makefile2
script5.sh
c2.o
hello1
script2.sh
.
script3.sh
c2.c
hello2
test.txt
c1.c
script1.sh
c3.c
Makefile1
```

```

zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ ./hello2 .
file1.txt
Makefile2
script5.sh
c2.o
hello1
script2.sh
.
script3.sh
c2.c
hello2
test.txt
c1.c
script1.sh
c3.c
Makefile1
..
Makefile3
script4.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ make -f Makefile2 clean
rm -rf *.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ mkdir testdir
zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ touch testdir/file1 testdir/file2 testdir/file3
zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ ./hello2 testdir
file2
.
file1
file3
..
zouxin@zouxin-virtual-machine:~/Desktop/B23040920\linx$ echo B23040920
B23040920

```

(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

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo B23040920
B23040920
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > c3.c << 'EOF'
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    char buf[1024] = {0};
    char buf2[1024] = {0};

    getcwd(buf, 1024);
    printf("Current directory: %s\n", buf);

    if(chdir("/home") < 0) {
        printf("error: cannot change to /home\n");
        return 1;
    }
    else {
        printf("success: changed to /home\n");
    }

    getcwd(buf2, 1024);
    printf("New directory: %s\n", buf2);

    return 0;
}
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ cat > Makefile3 << 'EOF'
hello3: c3.o
    gcc -o hello3 c3.o

c3.o: c3.c
    gcc -c c3.c

clean:
    rm -rf *.o
EOF
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ make -f Makefile3
gcc -c c3.c
gcc -o hello3 c3.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ls -l hello3 c3.o
-rw-rw-r-- 1 zouxin zouxin 2256 12月 28 23:08 c3.o
-rwxrwxr-x 1 zouxin zouxin 16136 12月 28 23:08 hello3
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./hello3
Current directory: /home/zouxin/Desktop/B23040920liunx
success: changed to /home
New directory: /home
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ make -f Makefile3 clean
rm -rf *.o
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ls -l *.o
ls: cannot access '*.o': No such file or directory
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./hello3
Current directory: /home/zouxin/Desktop/B23040920liunx
success: changed to /home
New directory: /home
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ pwd
/home/zouxin/Desktop/B23040920liunx
```