OS LAB 4: Unix System Calls

USAGE

- Open **Kernel/Terminal**
- Enter the command written in snippets

```
vi first_file
```

- Write anything
- press `esc` and type `:wq!` to save

```
cc project_name.c
```

```
/a.out
```

1. **Write a program to copy the contents of one file to another file using system calls**

CODE

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

int main()
{
  char buf;
  int fd_one, fd_two;

  fd_one = open("first_file", O_RDONLY);

  if (fd_one == -1)
  {
```

```
    printf("Error opening first_file\n");
    close(fd_one);
  }

  fd_two = open("second_file",
          O_WRONLY | O_CREAT,
          S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);

  while(read(fd_one, &buf, 1))
  {
    write(fd_two, &buf, 1);
  }

  printf("Successful copy");

  close(fd_one);
  close(fd_two);
}
```

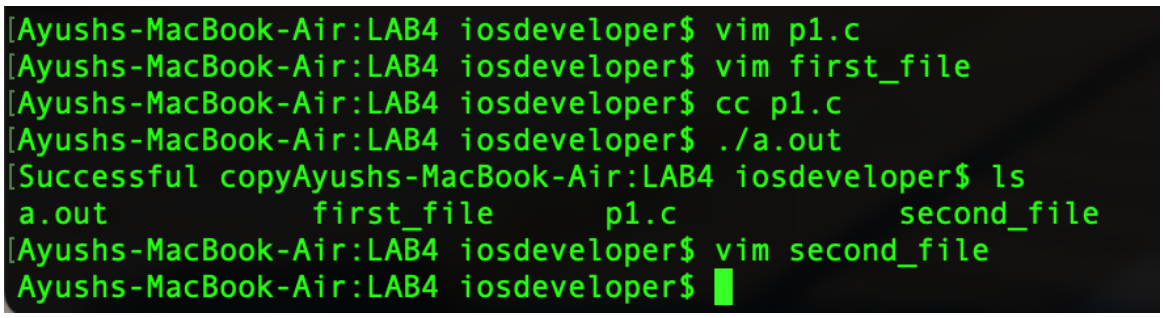
RESULT

first_file (before):

```
hello world
bye
```

second_file (after):

```
hello world
bye
```



2. **Write a program to append contents of one file to another file**

CODE

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

int main()
{
  char buf;
  int fd_one, fd_two;

  fd_one = open("first_file", O_RDONLY);

  if (fd_one == -1)
  {
    printf("Error opening first_file\n");
    close(fd_one);
  }

  fd_two = open("second_file",
          O_WRONLY | O_APPEND | O_CREAT);

  while(read(fd_one, &buf, 1))
  {
    write(fd_two, &buf, 1);
  }

  printf("Successful copy");

  close(fd_one);
  close(fd_two);
}
```

RESULT

first_file (before):

```
max
```

second_file (after append):

```
hello world
bye
max
```

3. **Write a program to reverse copy**

CODE

```
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>

int main() {

        int fd1, fd2;
        char c;
        int offset;

        fd1 = open("first_file", O_RDONLY);      //open file to read
        if (fd1 < 0) {
                printf("%s", "Open error");
        }

        fd2 = open("second_file", O_WRONLY | O_CREAT);
        if(fd2 < 0) {
                printf("%s", "Open error");
        }

        offset = lseek(fd1, 0, SEEK_END);    //go to the end of the file

        while (offset > 0) {
                read(fd1, &c, 1);
                write(fd2, &c, 1);
                lseek(fd1, -2, SEEK_CUR);
                offset--;
        }

        close(fd1);  //close the files
        close(fd2);

        return 0;
```

```
}
```

RESULT

first_file (before):

```
abcdefghijklmnopqrstuvwxyz
```

second_file (after lseek):

```
^@zyxwvutsrqponmlkjihgfedba
```

4. **Write a program to illustrate cat cmd**

CODE

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<fcntl.h>
int main( int argc,char *argv[3] )
{
int fd,i;
char buf[2];
fd=open(argv[1],O_RDONLY,0777);
if(fd==-argc)
{
printf("file open error");
}
else
{
while((i=read(fd,buf,1))>0)
{
printf("%c",buf[0]);
}
close(fd);
}
}
```

USAGE

```
-> cat > first_file "Hello World"
-> cc p4.c
-> ./a.out first_file
```

RESULT

```
Hello World
```