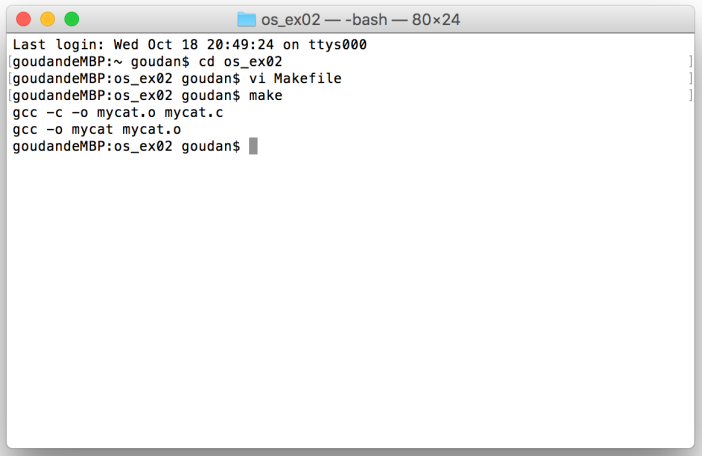
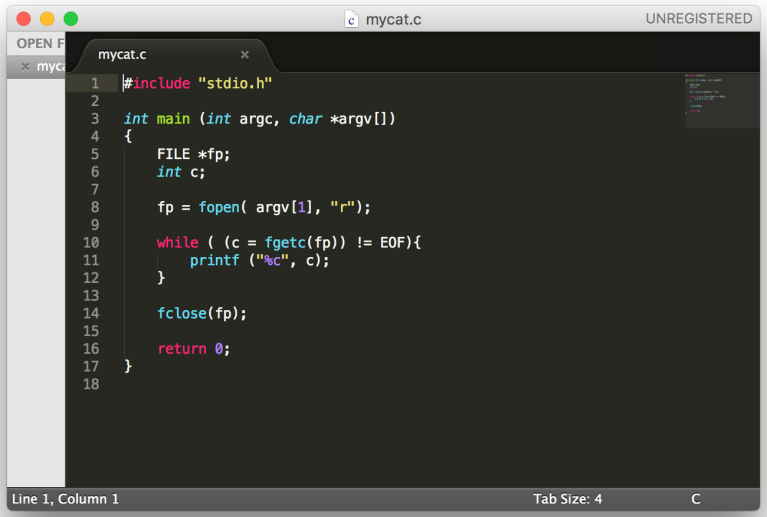
EX1:

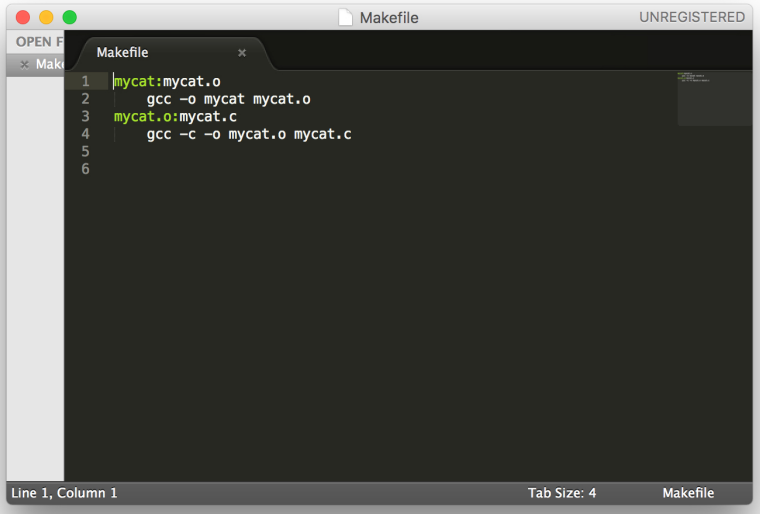
1.


os_ex02 — -bash — 80×24
Last login: Wed Oct 18 20:49:24 on ttys000
goudandeMBP:~ goudan$ cd os_ex02
goudandeMBP:os_ex02 goudan$ vi Makefile
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycat.o mycat.c
gcc -o mycat mycat.o
goudandeMBP:os_ex02 goudan$

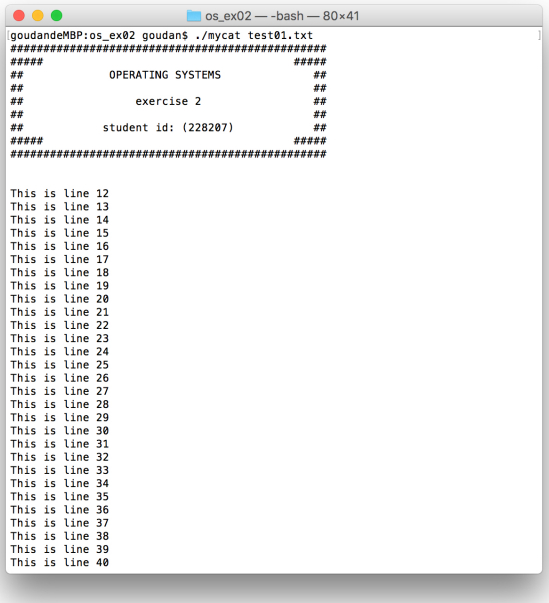
2.


mycat.c
UNREGISTERED
OPEN F
mycat.c
#include "stdio.h"
int main (int argc, char *argv[])
{
FILE *fp;
int c;
fp = fopen(argv[1], "r");
while ((c = fgetc(fp)) != EOF){
printf ("%c", c);
}
fclose(fp);
return 0;
}
Line 1, Column 1
Tab Size: 4
C

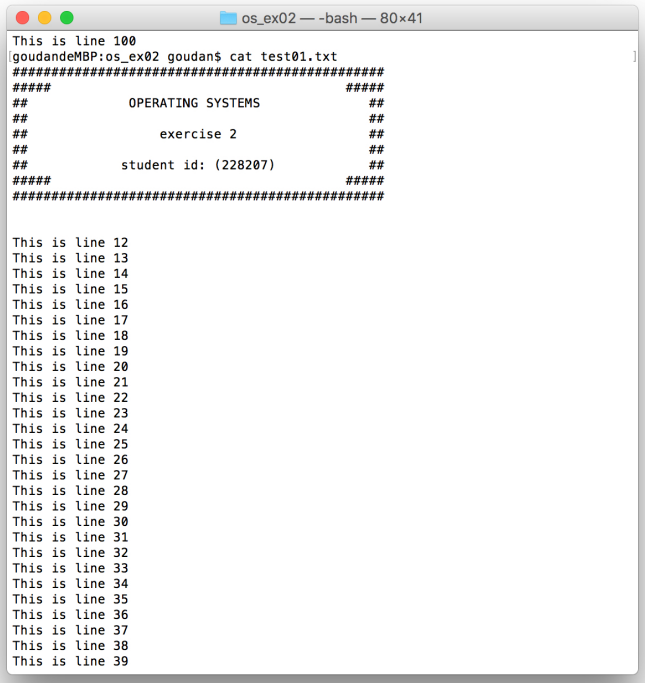
3.


Makefile
UNREGISTERED
OPEN F
Makefile
1 mycat:mycat.o
2 gcc -o mycat mycat.o
3 mycat.o:mycat.c
4 gcc -c -o mycat.o mycat.c
5
6
Line 1, Column 1
Tab Size: 4
Makefile

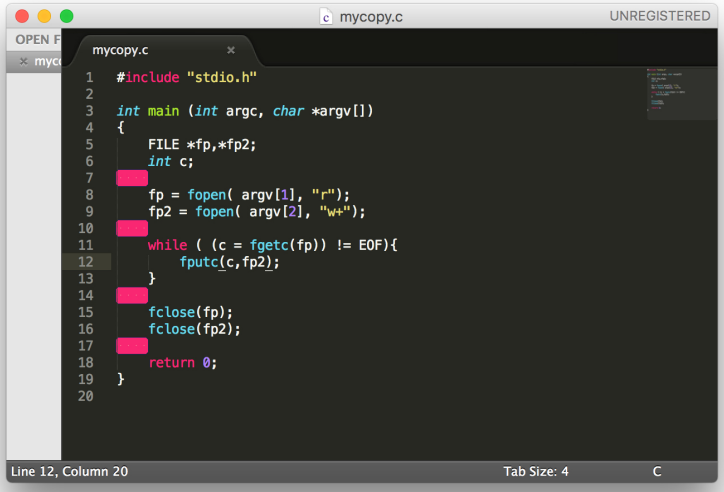
mycat:

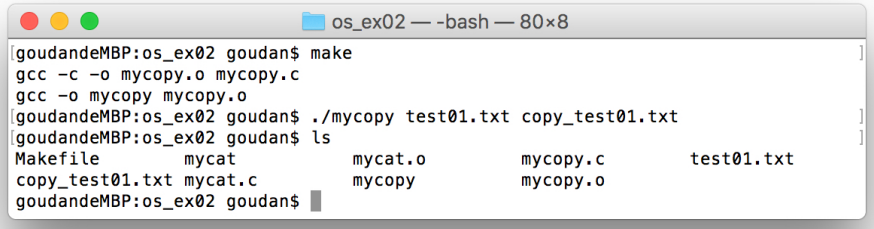

os_ex02 — -bash — 80×41
goudandeMBP:os_ex02 goudan$ ./mycat test01.txt
##
#####
OPERATING SYSTEMS
##
exercise 2
##
student id: (228207)
#####
##
This is line 12
This is line 13
This is line 14
This is line 15
This is line 16
This is line 17
This is line 18
This is line 19
This is line 20
This is line 21
This is line 22
This is line 23
This is line 24
This is line 25
This is line 26
This is line 27
This is line 28
This is line 29
This is line 30
This is line 31
This is line 32
This is line 33
This is line 34
This is line 35
This is line 36
This is line 37
This is line 38
This is line 39
This is line 40

cat:


os_ex02 — -bash — 80×41
This is line 100
goudandeMBP:os_ex02 goudan$ cat test01.txt
##
#####
OPERATING SYSTEMS
##
exercise 2
##
student id: (228207)
#####
##
This is line 12
This is line 13
This is line 14
This is line 15
This is line 16
This is line 17
This is line 18
This is line 19
This is line 20
This is line 21
This is line 22
This is line 23
This is line 24
This is line 25
This is line 26
This is line 27
This is line 28
This is line 29
This is line 30
This is line 31
This is line 32
This is line 33
This is line 34
This is line 35
This is line 36
This is line 37
This is line 38
This is line 39

4.

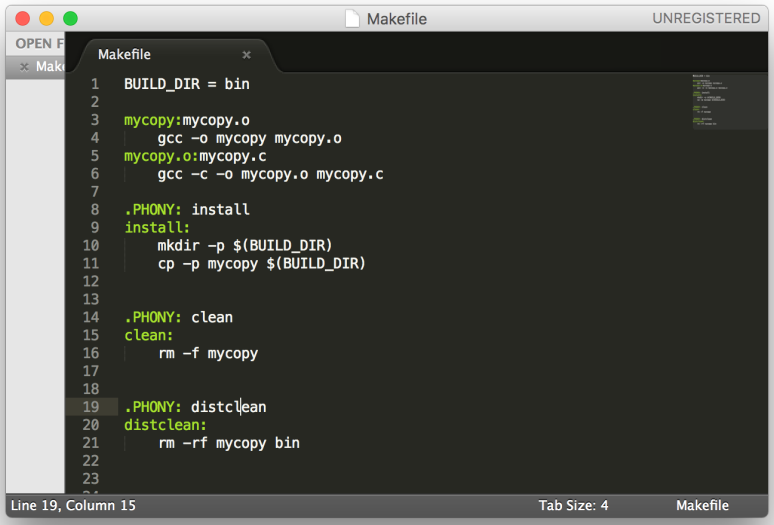

mycopy.c
UNREGISTERED
OPEN F
mycopy.c
#include "stdio.h"
int main (int argc, char *argv[])
{
FILE *fp,*fp2;
int c;
fp = fopen(argv[1], "r");
fp2 = fopen(argv[2], "w+");
while ((c = fgetc(fp)) != EOF){
fputc(c,fp2);
}
fclose(fp);
fclose(fp2);
return 0;
}
Line 12, Column 20
Tab Size: 4
C

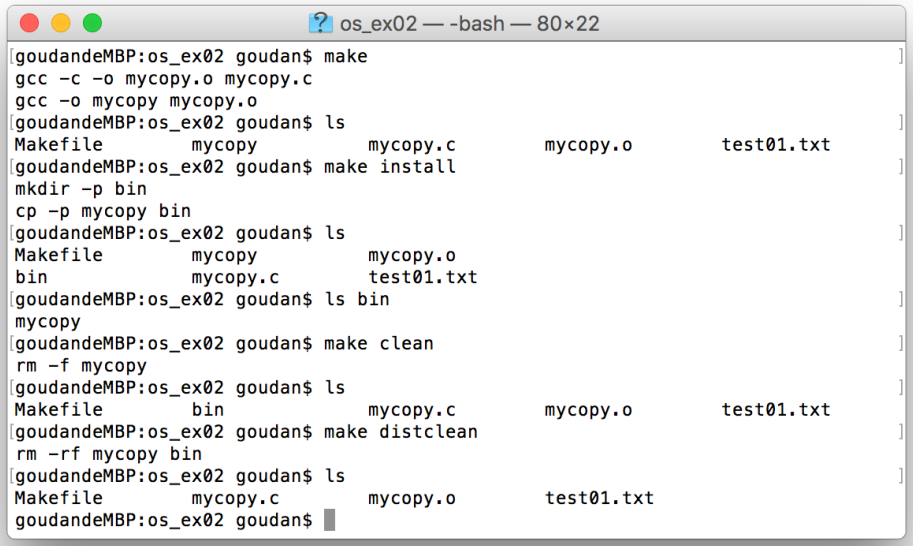


5.

A stream opened in text mode we get newline translation on non-*nix systems (it's also used for network communications, but this isn't supported by the standard library). In *nix newline is just ASCII linefeed, \n, both for internal and external representation of text. In Windows the external representation often uses a carriage return + linefeed pair, "CRLF" (ASCII codes 13 and 10), which is converted to a single \n on input, and conversely on output.

EX2


Makefile
UNREGISTERED
Makefile
1 BUILD_DIR = bin
2
3 mycopy:mycopy.o
4 gcc -o mycopy mycopy.o
5 mycopy.o:mycopy.c
6 gcc -c -o mycopy.o mycopy.c
7
8 .PHONY: install
9 install:
10 mkdir -p $(BUILD_DIR)
11 cp -p mycopy $(BUILD_DIR)
12
13
14 .PHONY: clean
15 clean:
16 rm -f mycopy
17
18
19 .PHONY: distclean
20 distclean:
21 rm -rf mycopy bin
22
23
Line 19, Column 15
Tab Size: 4
Makefile


os_ex02 — -bash — 80×22
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycopy.o mycopy.c
gcc -o mycopy mycopy.o
goudandeMBP:os_ex02 goudan$ ls
Makefile mycopy mycopy.c mycopy.o test01.txt
goudandeMBP:os_ex02 goudan$ make install
mkdir -p bin
cp -p mycopy bin
goudandeMBP:os_ex02 goudan$ ls
Makefile mycopy mycopy.o
bin mycopy.c test01.txt
goudandeMBP:os_ex02 goudan$ ls bin
mycopy
goudandeMBP:os_ex02 goudan$ make clean
rm -f mycopy
goudandeMBP:os_ex02 goudan$ ls
Makefile bin mycopy.c mycopy.o test01.txt
goudandeMBP:os_ex02 goudan$ make distclean
rm -rf mycopy bin
goudandeMBP:os_ex02 goudan$ ls
Makefile mycopy.c mycopy.o test01.txt
goudandeMBP:os_ex02 goudan$

EX3

1.

cp_binary.c

```c
int main (int argc, char *argv[])
{
    FILE * file1,*file2;

    file1 = fopen(argv[1],"rb");
    file2 = fopen(argv[2],"wb");
    if(!file1)
    {
        printf("open error\n");
        return;
    }
    char c;
    int index = 0;
    fseek(file1,0,SEEK_END);
    int length = ftell(file1);
    //printf("%d\n",length);
    if(!length)
        return;
    while(!fseek(file1,index,SEEK_SET))
    {
        fread(&c,1,1,file1);
        fwrite(&c,1,1,file2);
        if(index == length - 1)
        {
            break;
        }
        index++;
    }
    fclose(file1);
    fclose(file2);

    return 0;
}
```

Line 13, Column 19 — Spaces: 4 — C

os_ex02 — -bash — 80×37

```
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycopy.o mycopy.c
gcc -o mycopy mycopy.o
goudandeMBP:os_ex02 goudan$ ./mycopy binary.png copy.png
goudandeMBP:os_ex02 goudan$ ls
Makefile        copy.png        mycopy.c        test01.txt
binary.png      mycopy          mycopy.o
goudandeMBP:os_ex02 goudan$ file binary.png
binary.png: PNG image data, 604 x 468, 8-bit/color RGBA, non-interlaced
goudandeMBP:os_ex02 goudan$ file copy.png
copy.png: PNG image data, 604 x 468, 8-bit/color RGBA, non-interlaced
goudandeMBP:os_ex02 goudan$ ./mycopy test01.txt copy01.txt
goudandeMBP:os_ex02 goudan$ ls
Makefile        copy.png        mycopy          mycopy.o
binary.png      copy01.txt      mycopy.c        test01.txt
goudandeMBP:os_ex02 goudan$ file test01.txt
test01.txt: ASCII text
goudandeMBP:os_ex02 goudan$ file copy01.txt
copy01.txt: ASCII text
goudandeMBP:os_ex02 goudan$ cat copy01.txt
##################################################
#####                                        #####
##             OPERATING SYSTEMS                ##
##                                              ##
##                 exercise 2                   ##
##                                              ##
##            student id: (228207)              ##
#####                                        #####
##################################################


This is line 12
This is line 13
This is line 14
This is line 15
This is line 16
This is line 17
```

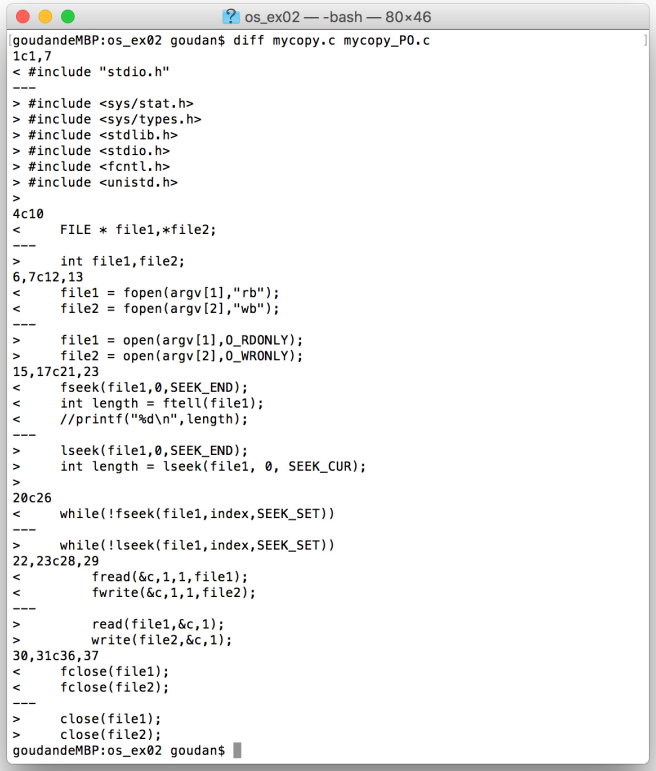
yes, it's possible to use this program to copy text files.

2.

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

int main (int argc, char *argv[])
{
    int file1,file2;

    file1 = open(argv[1],O_RDONLY);
    file2 = open(argv[2],O_WRONLY);
    if(!file1)
    {
        printf("open error\n");
        return 0;
    }
    char c;
    int index = 0;
    lseek(file1,0,SEEK_END);
    int length = lseek(file1, 0, SEEK_CUR);

    if(!length)
        return 0;
    while(!lseek(file1,index,SEEK_SET))
    {
        read(file1,&c,1);
        write(file2,&c,1);
        if(index == length - 1)
        {
            break;
        }
        index++;
    }
    close(file1);
    close(file2);

    return 0;
}
```

3.

```
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycopy_PO.o mycopy_PO.c
gcc -o mycopy_PO mycopy_PO.o
goudandeMBP:os_ex02 goudan$ ./mycopy_PO binary.png copy.png
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycopy_PO.o mycopy_PO.c
gcc -o mycopy_PO mycopy_PO.o
goudandeMBP:os_ex02 goudan$ ./mycopy_PO binary.png copy.png
goudandeMBP:os_ex02 goudan$ make
gcc -c -o mycopy_PO.o mycopy_PO.c
gcc -o mycopy_PO mycopy_PO.o
goudandeMBP:os_ex02 goudan$ ./mycopy_PO binary.png copy.png
goudandeMBP:os_ex02 goudan$ ls
Makefile        copy.png        mycopy_PO       mycopy_PO.o
binary.png      mycopy.c        mycopy_PO.c     test01.txt
goudandeMBP:os_ex02 goudan$ stat copy.png
16777220 89817373 -rw-r--r-- 1 goudan staff 0 48762 "Oct 19 00:44:39 2017" "Sep
29 16:00:59 2017" "Oct 19 00:44:32 2017" "Sep 29 16:00:59 2017" 4096 96 0 copy.p
ng
goudandeMBP:os_ex02 goudan$ stat binary.png
16777220 88790458 -rw-r--r-- 1 goudan staff 0 48762 "Oct 19 00:44:36 2017" "Sep
29 16:00:59 2017" "Sep 29 16:01:00 2017" "Sep 29 16:00:59 2017" 4096 96 0 binary
.png
goudandeMBP:os_ex02 goudan$
```


os_ex02 — -bash — 80×46
goudandeMBP:os_ex02 goudan$ diff mycopy.c mycopy_PO.c
1c1,7
< #include "stdio.h"

> #include <sys/stat.h>
> #include <sys/types.h>
> #include <stdlib.h>
> #include <stdio.h>
> #include <fcntl.h>
> #include <unistd.h>
>
4c10
<   FILE * file1,*file2;

>   int file1,file2;
6,7c12,13
<   file1 = fopen(argv[1],"rb");
<   file2 = fopen(argv[2],"wb");

>   file1 = open(argv[1],O_RDONLY);
>   file2 = open(argv[2],O_WRONLY);
15,17c21,23
<   fseek(file1,0,SEEK_END);
<   int length = ftell(file1);
<   //printf("%d\n",length);

>   lseek(file1,0,SEEK_END);
>   int length = lseek(file1, 0, SEEK_CUR);
>
20c26
<   while(!fseek(file1,index,SEEK_SET))

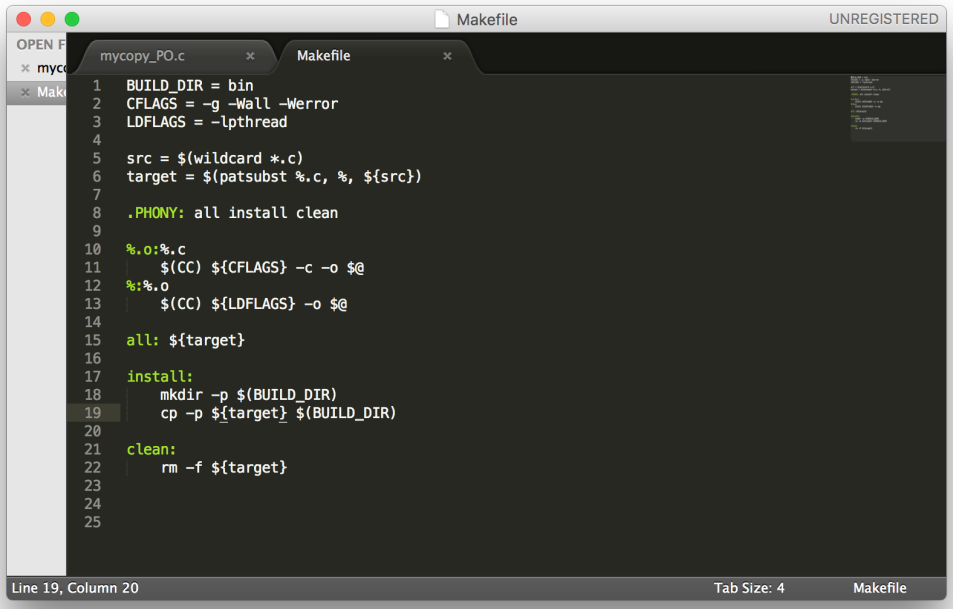
>   while(!lseek(file1,index,SEEK_SET))
22,23c28,29
<   fread(&c,1,1,file1);
<   fwrite(&c,1,1,file2);

>   read(file1,&c,1);
>   write(file2,&c,1);
30,31c36,37
<   fclose(file1);
<   fclose(file2);

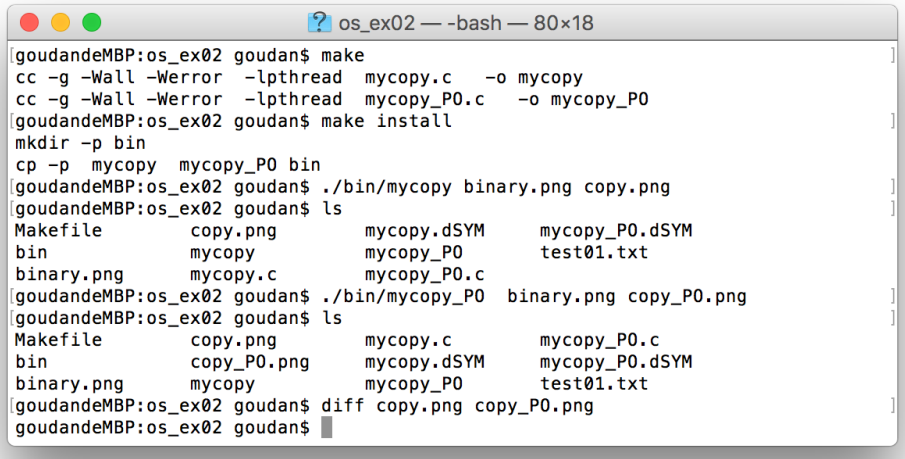
>   close(file1);
>   close(file2);
goudandeMBP:os_ex02 goudan$

EX4

1.

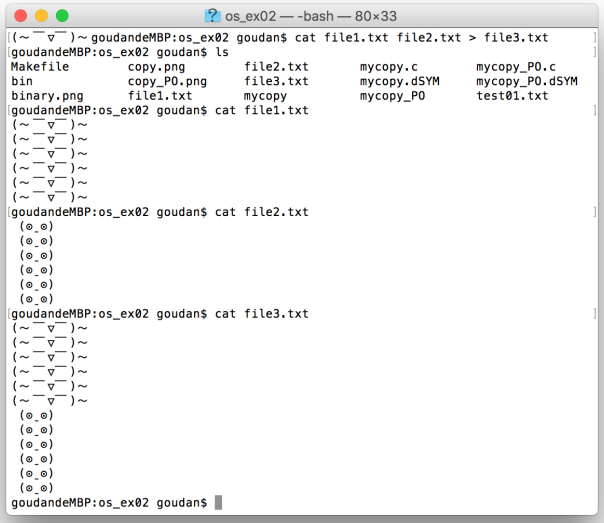

Makefile
UNREGISTERED
mycopy_PO.c
Makefile
BUILD_DIR = bin
CFLAGS = -g -Wall -Werror
LDFLAGS = -lpthread
src = $(wildcard *.c)
target = $(patsubst %.c, %, ${src})
.PHONY: all install clean
%.o:%.c
$(CC) ${CFLAGS} -c -o $@
%:%.o
$(CC) ${LDFLAGS} -o $@
all: ${target}
install:
mkdir -p $(BUILD_DIR)
cp -p ${target} $(BUILD_DIR)
clean:
rm -f ${target}
Line 19, Column 20
Tab Size: 4
Makefile

2.

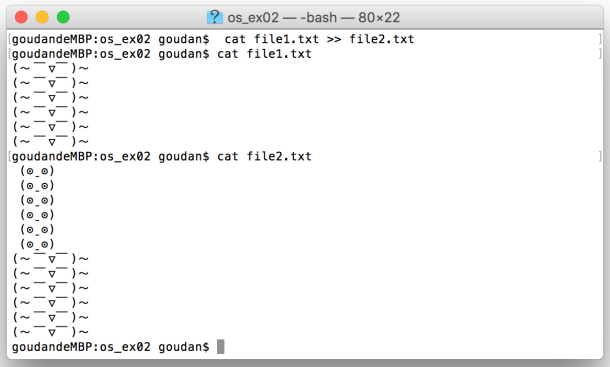

os_ex02 — -bash — 80×18
goudandeMBP:os_ex02 goudan$ make
cc -g -Wall -Werror -lpthread mycopy.c -o mycopy
cc -g -Wall -Werror -lpthread mycopy_PO.c -o mycopy_PO
goudandeMBP:os_ex02 goudan$ make install
mkdir -p bin
cp -p mycopy mycopy_PO bin
goudandeMBP:os_ex02 goudan$ ./bin/mycopy binary.png copy.png
goudandeMBP:os_ex02 goudan$ ls
Makefile copy.png mycopy.dSYM mycopy_PO.dSYM
bin mycopy mycopy_PO test01.txt
binary.png mycopy.c mycopy_PO.c
goudandeMBP:os_ex02 goudan$ ./bin/mycopy_PO binary.png copy_PO.png
goudandeMBP:os_ex02 goudan$ ls
Makefile copy.png mycopy.c mycopy_PO.c
bin copy_PO.png mycopy.dSYM mycopy_PO.dSYM
binary.png mycopy mycopy_PO test01.txt
goudandeMBP:os_ex02 goudan$ diff copy.png copy_PO.png
goudandeMBP:os_ex02 goudan$

EX5

1.

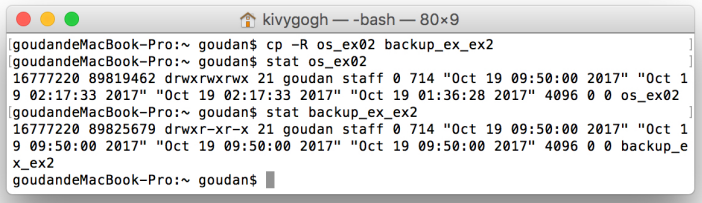

os_ex02 — -bash — 80×33
(~ ¯▽¯)~ goudandeMBP:os_ex02 goudan$ cat file1.txt file2.txt > file3.txt
goudandeMBP:os_ex02 goudan$ ls
Makefile copy.png file2.txt mycopy.c mycopy_PO.c
bin copy_PO.png file3.txt mycopy.dSYM mycopy_PO.dSYM
binary.png file1.txt mycopy mycopy_PO test01.txt
goudandeMBP:os_ex02 goudan$ cat file1.txt
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
goudandeMBP:os_ex02 goudan$ cat file2.txt
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
goudandeMBP:os_ex02 goudan$ cat file3.txt
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
goudandeMBP:os_ex02 goudan$

2.

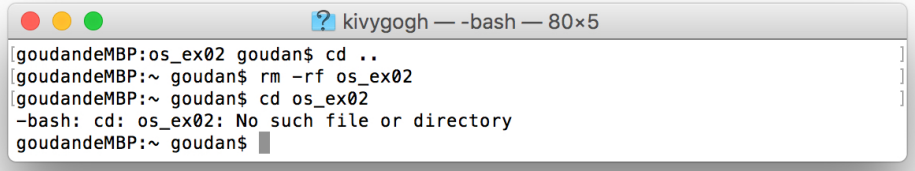

os_ex02 — -bash — 80×22
goudandeMBP:os_ex02 goudan$ cat file1.txt >> file2.txt
goudandeMBP:os_ex02 goudan$ cat file1.txt
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
goudandeMBP:os_ex02 goudan$ cat file2.txt
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
(o_o)
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
(~ ¯▽¯)~
goudandeMBP:os_ex02 goudan$

EX6

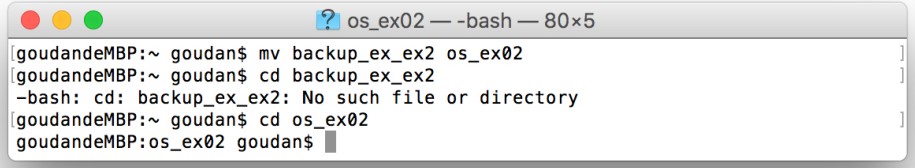
1.


kivygogh — -bash — 80×9
goudandeMacBook-Pro:~ goudan$ cp -R os_ex02 backup_ex_ex2
goudandeMacBook-Pro:~ goudan$ stat os_ex02
16777220 89819462 drwxrwxrwx 21 goudan staff 0 714 "Oct 19 09:50:00 2017" "Oct 19 02:17:33 2017" "Oct 19 02:17:33 2017" "Oct 19 01:36:28 2017" 4096 0 0 os_ex02
goudandeMacBook-Pro:~ goudan$ stat backup_ex_ex2
16777220 89825679 drwxr-xr-x 21 goudan staff 0 714 "Oct 19 09:50:00 2017" "Oct 19 09:50:00 2017" "Oct 19 09:50:00 2017" 4096 0 0 backup_ex_ex2
goudandeMacBook-Pro:~ goudan$

1.

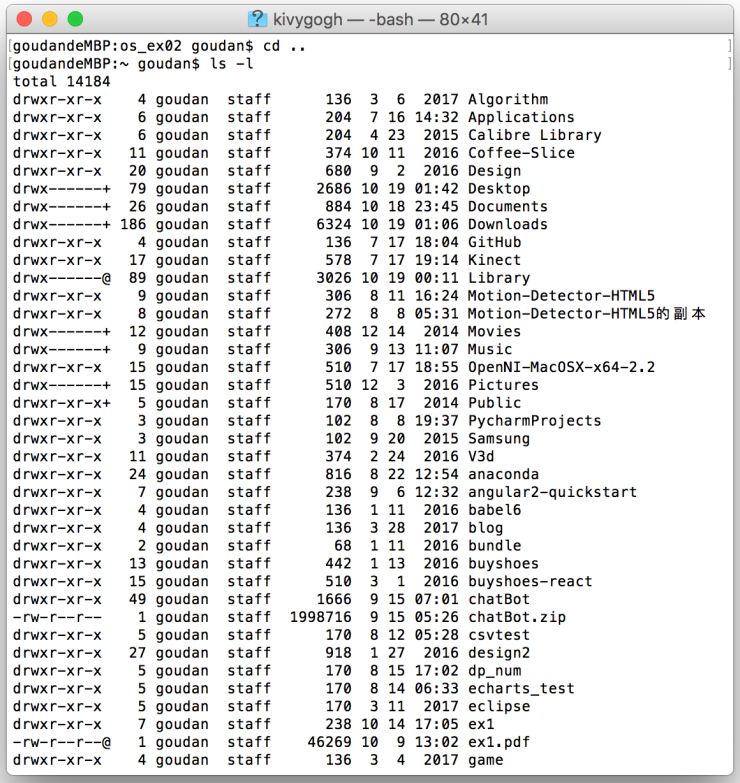

kivygogh — -bash — 80×5
goudandeMBP:os_ex02 goudan$ cd ..
goudandeMBP:~ goudan$ rm -rf os_ex02
goudandeMBP:~ goudan$ cd os_ex02
-bash: cd: os_ex02: No such file or directory
goudandeMBP:~ goudan$

2.

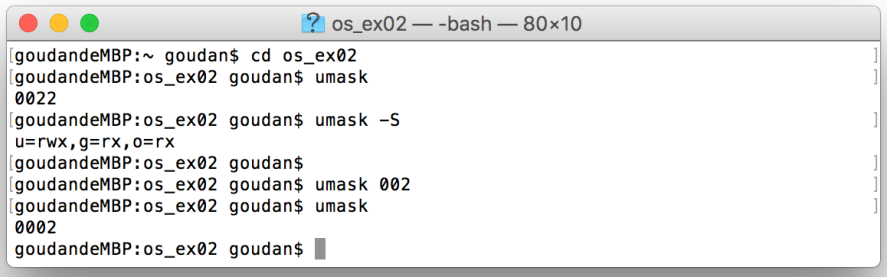

os_ex02 — -bash — 80×5
goudandeMBP:~ goudan$ mv backup_ex_ex2 os_ex02
goudandeMBP:~ goudan$ cd backup_ex_ex2
-bash: cd: backup_ex_ex2: No such file or directory
goudandeMBP:~ goudan$ cd os_ex02
goudandeMBP:os_ex02 goudan$

EX7

1.



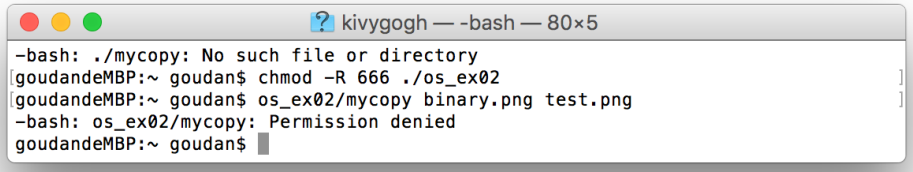
2.



The umask acts as a set of permissions that applications cannot set on files. It's a file mode creation mask for processes and cannot be set for directories

itself. Most applications would not create files with execute permissions set, so they would have a default of 666, which is then modified by the umask.

3.



4.