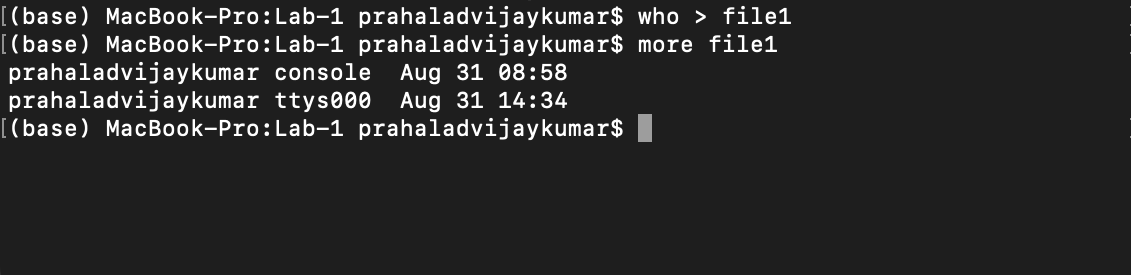
**LAB – 1 (Part-B)**

1. who > file1

more file1

Result:

prahaladvijaykumar console Aug 31 08:58

prahaladvijaykumar ttys000 Aug 31 14:34

1. Code:

question2.sh

#!usr/bin/env bash

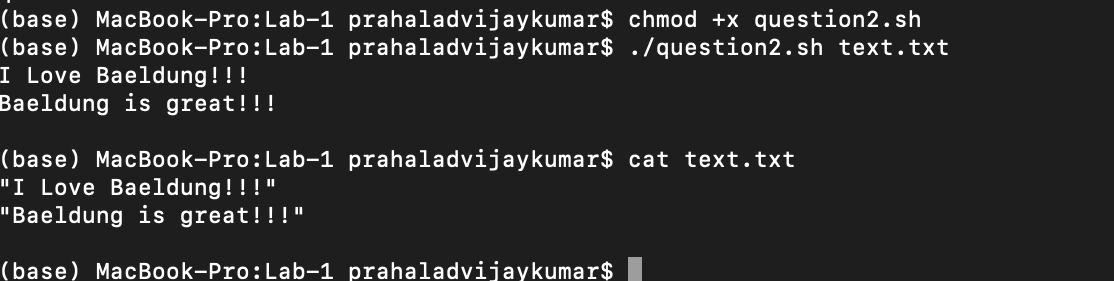
#accepting input of filename and reading it in variable file

#using replace command of sed s

#replace ^. to ' '(^. represents the first character after a new line)

#replace .$ to ' '(.$ represents the last character in a line)

sed 's/^.//;s/.$//' $1

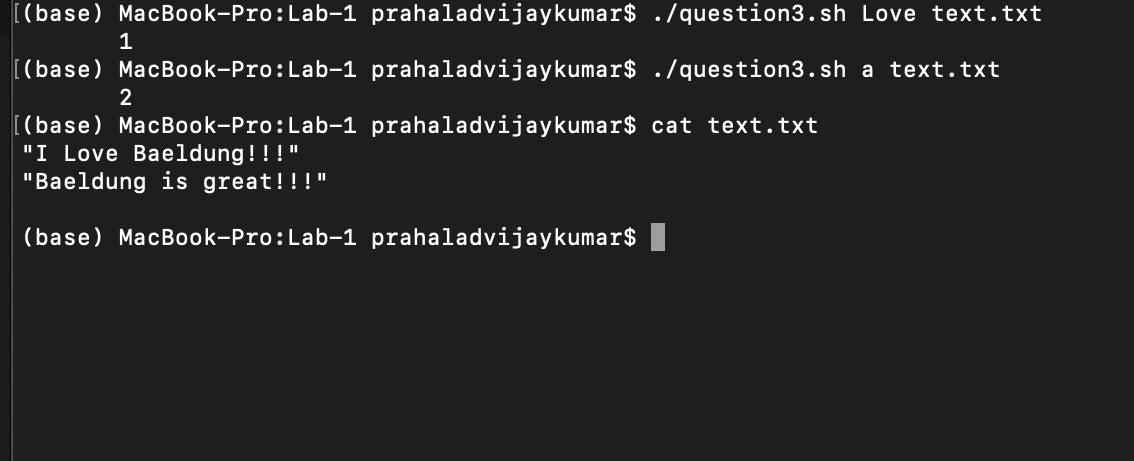


1. Code

#!usr/bin/env bash

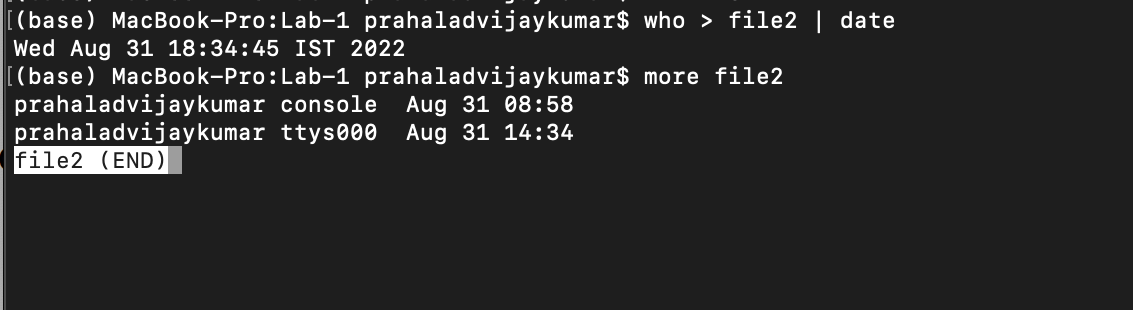
grep -h -i $1 $2 | wc -l

#grep-h displays number of matching lines and wc -l counts the number of matching lines



1. who > myfile2 | date

(I created file2 instead fo myfile2)





C:

#include <stdio.h>

int main(){

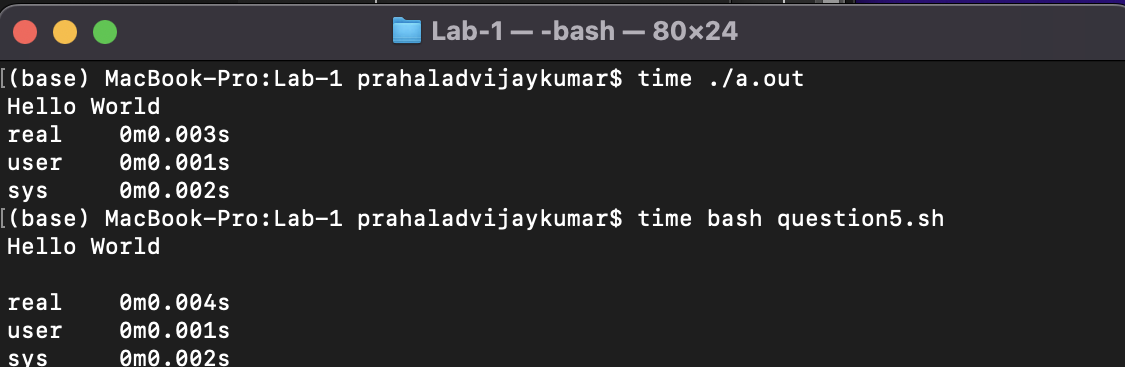
printf("Hello World");

return 0;

}

Bash:

echo “Hello World”



1. Code:

for i; do

if [ ! -f $i ]

then

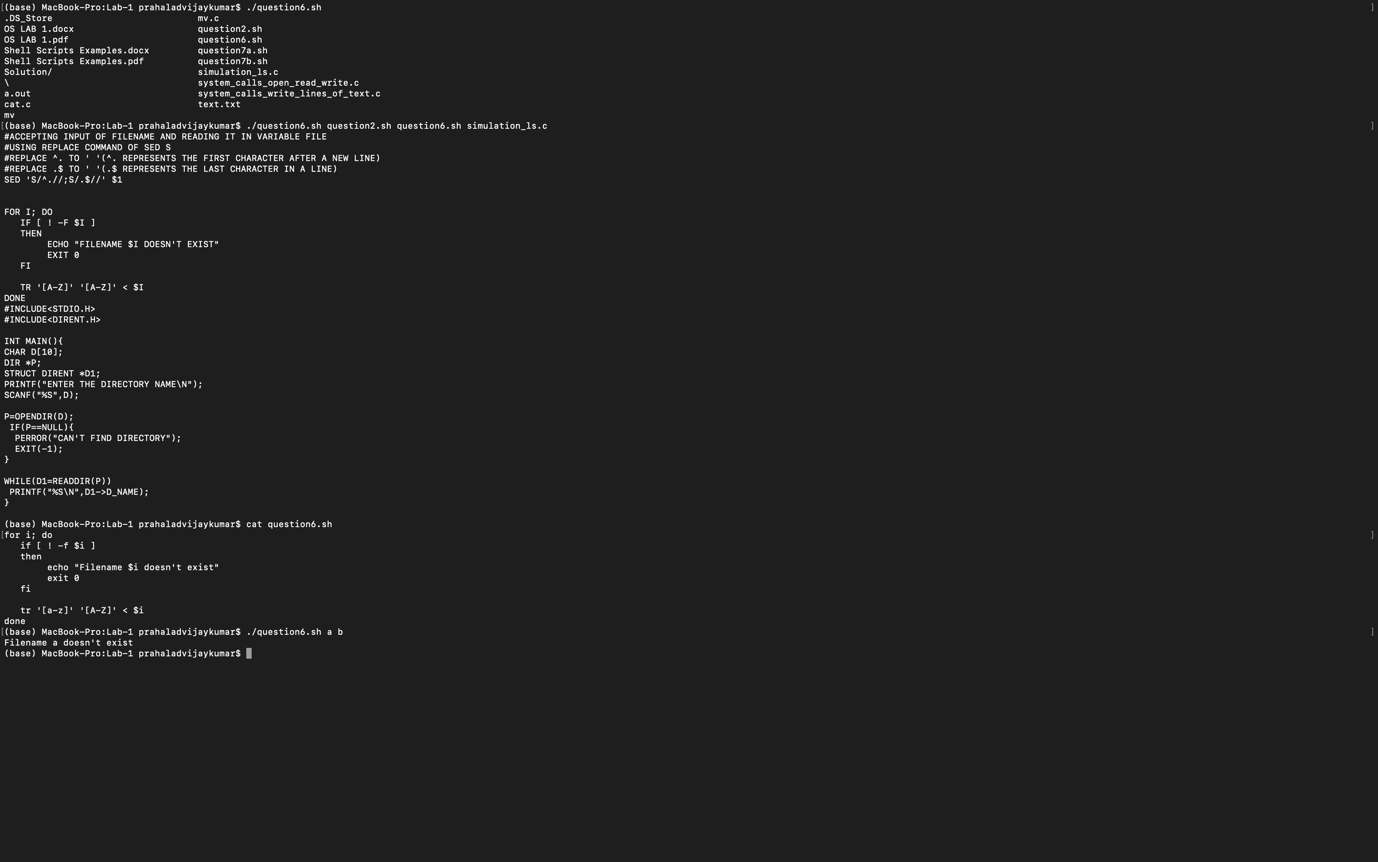
echo "Filename $i doesn't exist"

exit 0

fi

tr '[a-z]' '[A-Z]' < $i

done



1. (a)

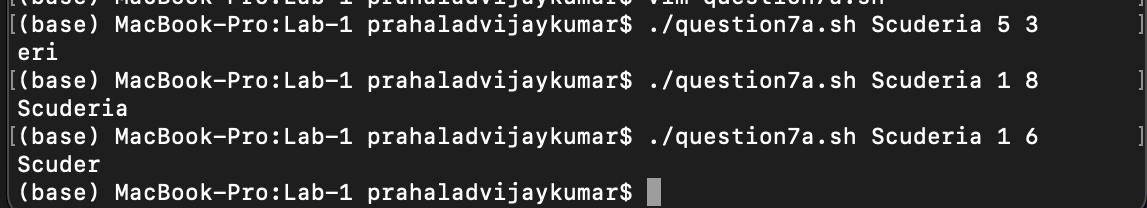
#!usr/bin/env bash

#input the first index and the length of the substring

var=(`expr $3 + $2 - 1`)

cut -c $2-$var <<< $1

#returns the substring



(b)

#!usr/bin/env bash

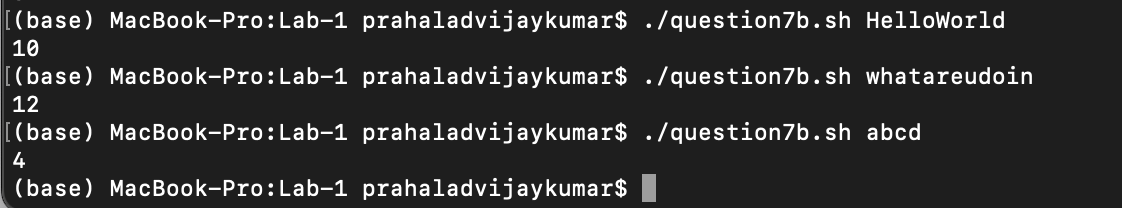
str=$1

ans=`echo $str | wc -c `

#counts the space which seperates ./<filename> <string>

#return length+1 so subtract 1 to obtain the result

echo $((ans -1))



The above code doesn’t recognize spaces but it can be done by checking $2 ,$3 .. as spaces increase)

1. a)cat command

#include<stdio.h>

#include <fcntl.h>

#include <unistd.h>

int main(int argc,char \*\*argv){

if(argc==1){

printf("Expected File Names\n");

return -1;

}

char content[2000000];

int fd,n;

for(int i=1;i<argc;i++){

fd = open(argv[i],O\_RDONLY);

if(fd==-1){

printf("\nError:Valid FileName Required\n");

return -1;

}

else{

n = read(fd,content,2000000);

n = write(1,content,n);

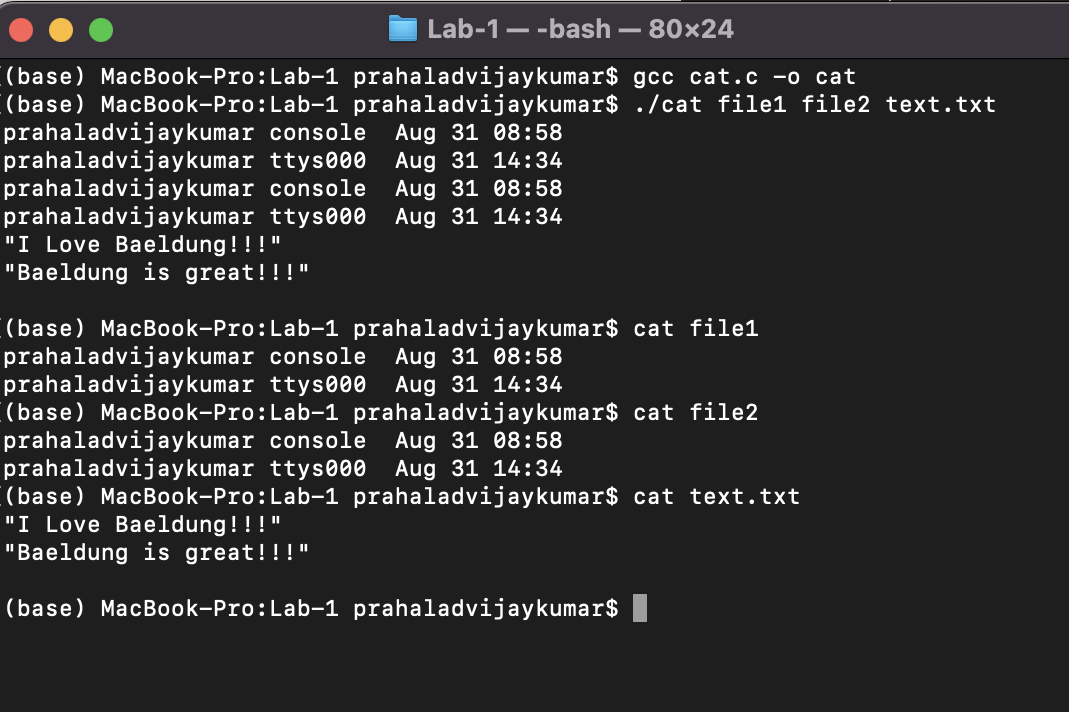
}

close(fd);

}

return 0;

}

It can take any number of file as input and expects a file that exist in the folder . it prints the text in the file 

b)Considered that the file exist in the same directory which is being accessed in linux(because doing the other will get complicated . or can be done using reading that file and writing on the file) and end the folder section with a (/)

#include <stdio.h>

#include <dirent.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#include <fcntl.h>

int main(int argc,char \*\*argv){

if(argc<3){

printf("Parameters aren't enough\n");

return -1;

}

DIR \*p;

p=opendir(argv[argc-1]);

if(p==NULL){

perror("can't find directory");

return -1;

}

int fd;

for(int i=1;i<argc-1;i++){

fd = open(argv[i],O\_RDONLY);

if(fd==-1){

printf("\nError:Valid FileName Required\n");

return -1;

}

else{

char \*newfile;

newfile=malloc(strlen(argv[i])+strlen(argv[argc-1])+1);

strcpy(newfile,argv[argc-1]);

strcat(newfile,argv[i]);

if(rename(argv[1],newfile)==-1){

printf("Syntax Error .");

}

else{

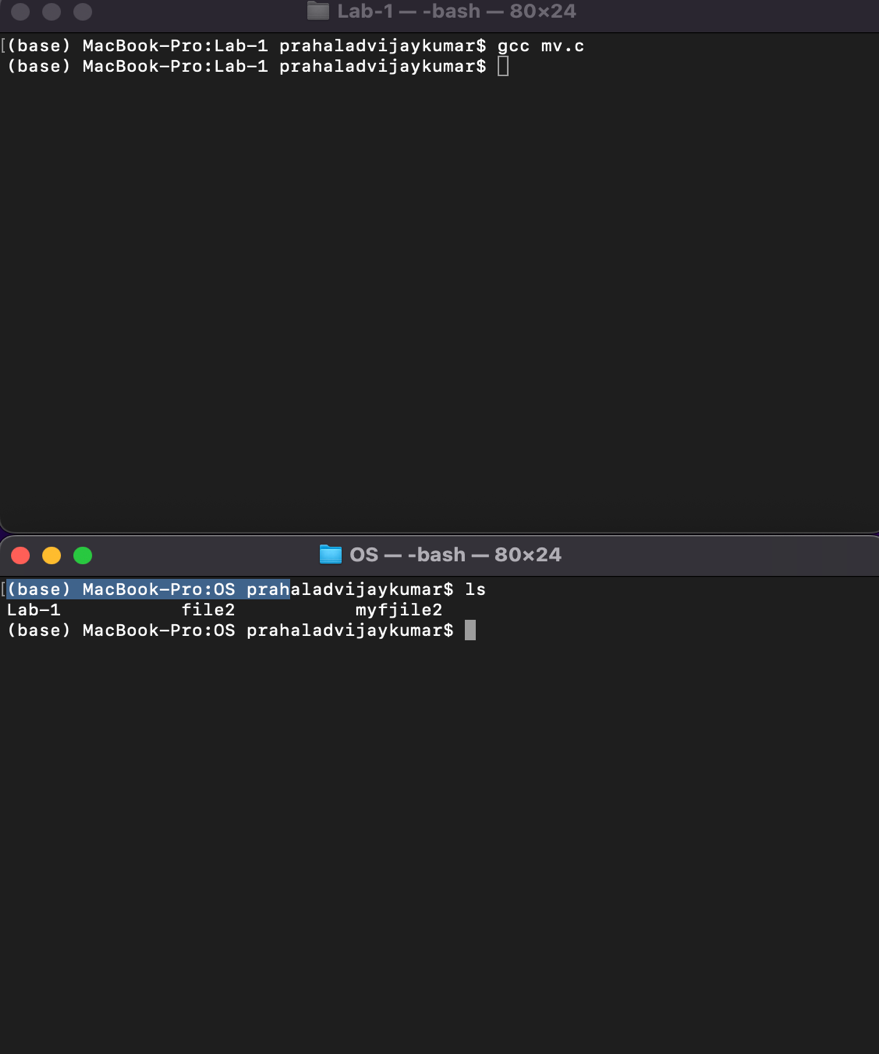
;

}

}

}

}

Text

Description automatically generated