## OPERATING SYSTEM

## (REPORT PROGRAMMING PART)

Name: MUNIR JAWAD

Matr no: 158385

Exam date: 03 September 2012.

## Bash :

You are requested to perform text manipulation (using bash build in commands or common unix commands) on a C program Language C source file (main.c) you have to

1. Print to console the number of lines in the file ?
2. Print to console how many lines contains the C keyword “while” ?
3. Print to console how many blank lines in the file ?
4. Save in another file (raw\_main.c)the contents of the source file removing at the blank lines?
5. Print to console only the lines that donot begins with one or more blank characters
6. Print to console that are terminated a C comment (i.e by the “\*/” sequence)?

## Solution :

Now by testing everything at home I come up with the solution as follows

1. wc –l main.c

now this is simple solution as we need just the no of lines in the file, so above command give us so.

1. For this solution we had to use the grep, as follows

grep –c .\*while.\* main.c

1. For this we would again use grep as :

grep –c ^$ main.c

1. We could do this as follow. Dear sir I was little bit confuse in this one, but I do it write now , and saved in another file raw\_main.c content s of main.c removing blank spaces , so the solution is given as

grep -v ^$ main.c 1>raw\_main.c

1. Now there was a trick here, not we only need to see the first blank space in each line, not for more than two blank spaces, as the condition is that no single blank character. My bas code is as follows :

grep -v ^’ ‘ main.c

1. grep ‘\*/’ main.c

Dear sir , In doing bash at home , I created a file main.c and fill in the file with the contents and started testing my bash script. I have also created a file raw\_main.c in order to copy the contents of main.c in raw\_main.c in exercise no# 4. I hope that you will like my solutions. I hope you will not consider silly mistakes for the exam.

## Problem 2:

/\* write a C program:

a) A father must fork three sons(SonA, SonB, SonC)

b) SonA sends to SonB a string using pipe and exit.

c) SonB check if palindrome or pass to sonC.

d) SonC not able to recieve nothing from input in 3 seconds he exits

e) SonC after getting string has to reverse it.

f) SonB prints both original and the reversed one.and exit

g) father process wait for three children before exit.\*/

The solution that is in red, is the things that differ or is added new to the exam that was done in class. I hope the professor will very kind.

## Solution:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<sys/wait.h>

#include<sys/types.h>

#include<sys/errno.h>

#include<unistd.h>

#include<sys/ipc.h>

#include<signal.h> // used for handler.

#include<time.h>

#define MAXLINE 100

#define N 30

// father process

void handler(int signum )// used for wait of 3 seconds.

{

kill(getpid(),SIGKILL);

}

int main()

{

pid\_t pid\_1,pid\_2,pid\_3;// for SonA, SonB, SonC

int check=0;// add a check to tell if string is palindrome or not.

int i,j,temp;

char revstr[N];

char string[N]="civic";// i took this one particular example where i used a

palindrome word civic to test my C program.

char b[N],revstring[N];

int status,ret,ret2,ret3,mypipe1[2],mypipe2[2],mypipe3[2];

pipe(mypipe1);

pipe(mypipe2);

pipe(mypipe3);

// for sonA

pid\_1=fork();

if(pid\_1==0)

{

// only for sonA nothing special for father in this.

write(mypipe1[1],string,strlen(string)+1);

exit(0);// sonA exit okk

}

// sonB

pid\_2=fork();

if(pid\_2==0)

{

// only sonB

ret=read(mypipe1[0],string,N);

// Algorithm to find the palindrome.

strcpy(b,string);// copy from string to b

// reverse the string by taking care of null character.

//Strrev(b, string);

for(i=strlen(b)-1,j=0;i>=0;i--,j++)

{

revstr[j]=b[i];

}

printf("reversed");

revstr[j]='\0';

if(strcmp(string,revstr)==0)// the string is palindrome

{

check++; // if palindrome do check++

}

// the string is palindrome

if(check>0)

{

printf("Palindrom string recieved\n");

// print if palindrome

}

else

{

// not a palindrome

// SonB sends string to SonC using a pipe

write(mypipe2[1],string,strlen(string)+1);

read(mypipe3[0],revstring,N);

// printf("%s no2 \n",revstring);

}

//printf("%s no1\n",string);

exit(0); // we terminated sonB

}

// fork SonC

pid\_3=fork();

if(pid\_3==0)

{ // sonC starts

signal(sigalrm,handler);// register the signal handler for alarm signal.

alarm(3);

ret2=read(mypipe2[0],string,N);

if(ret==-1)

exit(0);

else

{// reverse the string

revstring[ret2]=0;

j=strlen(revstring)-1;

i=0;

while(i<j)

{

temp=revstring[i];

revstring[i++]=revstring[j];

revstring[j--]=temp;

}

write(mypipe3[1],revstring,strlen(revstring)+1);

exit(0);

}

} // sonC finished

// father Process agian.

waitpid(pid\_1,&status,0);// to show that father dont die before sonA

waitpid(pid\_3,&status,0);// to show that father dont die before sonC

waitpid(pid\_2,&status,0);// to show that father dont die before sonB

exit(0);

}

REPORT:

Now I started my program by including in all the headers needed. Specially the string.h and signal.h and rest all headers are also included.

I declare as a global variable MAXLINE and N as 100 and 30 respectively.

I have used a handler function, used for the wait of 3 seconds, that in part e) of the above question we needed that IF sonC is not able to receive nothing from its input pipe in 3 seconds , he exits.

Now starts the main function:

After the initialization of the variable that I used. I will directly go on explaining how I perform the required task given in problem 2 of the exam. I started by a particular example , putting inside the char string[] a palindrome. And than write the algorithm.

I used three pipes i.e mypipe1 , mypipe2, and mypipe3.

From SonA to SonB we send the string using mypipe1 and than from SonB to SonC the string is sended using mypipe2 , and finally from SonC to SonB , mypipe3 is used to send the string.

IN the class exam I used only two pipes as treating mypipe2 to be bidirectional communicator, but during doing the program at home in Ubuntu, I have realized that that sending string in both direction through the same pipe creates problem, and gives a chance of error as SonB writes to SonC , if at the same time SonC writes to SonB than it could be a problem, or if sonC writes some garbage to sonB without even having anything from SonB.

Pid\_1= fork() ; creates SonA .

And sonA was created successfully; now SonA writes to SonB through mypipe1 and than exit.

Pid\_2=fork()

Creates SonB ,

Now the function needed to be performed by SonB are as follows:

To check if the string is palindrome, now I used the following algorithm in order to find if the string is palindrome or not:

Using the strcpy() in string.h ,I copied the string to temp string (in this case char b[]) , and than I reversed the string by taking care of null character, Now in class exam I used the function strrev() in string.h as we do for normal C codes. But when doing the program at home I have come to realize that strrev() was not functioning in ubuntu. So I had to reverse the string by the algorithm mentioned above. And in the end when I have found the reverse string, I used a string.h function i.e strcmp and compare the original string with the reverse string, and if the compare is ok we print to the console the string read is a palindrome. Now we used a check to do so.

if(strcmp(string,revstr)==0)// the string is palindrome

{

check++; // if palindrome do check++

}

Now if the check is greater than zero than surely our string read was a palindrome, otherwise we go the else statement.

Now if the string is not a palindrome, SonB writes to SonC using mypipe2. And start anticipating for the SonC to write the reverse string back to SonA.

And than we exit sonB.

Pid\_3=fork : Creates SOnC

SonC is successfully created

SonC has to perform as : if SonC is not able to receive nothing from its input pipe in 3 seconds , he exits. : now I have done this by using the

void handler(int signum )// used for wait of 3 seconds.

{

kill(getpid(),SIGKILL);

}

And by giving the command :

signal(sigalrm,handler);// register the signal handler for alarm signal.

alarm(3);

Now SonC read the string, reverse it and write it to SonB using mypipe3.the algorithm that I used for reverse string is mentioned in the program.

Now in the class exam I didn’t mentioned the handler, and I know it’s a mistake and I am really sorry for not doing it.

Now after writing to SonB , the process SOnB exit.

Now father process waits for the three children before exit, as the father can exit before the child. So I did it by:

// father Process agian.

waitpid(pid\_1,&status,0);// to show that father dont die before sonA

waitpid(pid\_3,&status,0);// to show that father dont die before sonC

waitpid(pid\_2,&status,0);// to show that father dont die before sonB

And after that we exit.

Dear sir I have really worked for the report. I hope you will be very kind in marking the exam.