Sir Syed CASE Institute of Technology

Muhammad Talha Ramzan
2330-0141
BS AI

Task no 01

Write a program where the child process sends an integer number to the parent process using an unnamed pipe. The parent process reads the number from the pipe and prints it,

Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
 int main(){
  int pipefd[2];
  pid_t pid;
 int num =42;

if(pipe(pipefd) == -1){
 perror("ppipe failed");
exit (1);
}
pid = fork();
if (pid<0){
perror ("fork failed");
exit(1);
}
if (pid == 0){
close (pipefd[0]);
```

```
if (pid<0){
perror ("fork failed");
exit(1);
}
if (pid == 0){
close (pipefd[0]);
printf("child: sending number %d to parent .\n",num);
write(pipefd[1], &num, sizeof(num));
close(pipefd[1]);
}
else{
close(pipefd[1]);
read(pipefd[0], &num, sizeof(num));
printf("parent: recived number %d fromchild.\n",num);
close(pipefd[0]);
}
return 0;
}
```

Out put

```
ubuntu@ubuntu:~/Desktop/talha$ gcc lab4.c -o lab4
ubuntu@ubuntu:~/Desktop/talha$ ./lab4
child: sending number 42 to parent .
parent: recived number 42 fromchild.
ubuntu@ubuntu:~/Desktop/talha$
```

Task no 02

Write a program where the parent process sends two integers to the child process using an unnamed
pipe. The child process reads the two numbers, calculates their sum, and prints the result

code

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

int main(){
int fd[2];
int num1 = 10, num2 = 20;
pid_t pid;
if(pid < 0){
perror("fork failed");
return 1;
}
if (pid > 0){
close (fd[0]);
write(fd[1], &num1, sizeof(num1));
write(fd[1], &num2, sizeof(num2));
close(fd[1]);
}
else{
int received1, received2, sum;
```

```
else{
int received1, received2, sum;
close(fd[1]);
read(fd[0], &received1, sizeof(received1));
read(fd[0], &received2, sizeof(received2));
close(fd[0]);
sum = received1 + received2;
printf("child: Received numbers  %d and %d. sum =%d\n", received1, received2, s
}
return 0;
}
```

Out put

```
ubuntu@ubuntu:~/Desktop$ gcc lab42.c -o talhabb
ubuntu@ubuntu:~/Desktop$ ./talhabb
fork failed: Success
ubuntu@ubuntu:~/Desktop$
```

Task no 03

Write a program to calculate factorial of a number. Following constraints must be fulfilled:

- Use pipe to solve your problem.
- Child process should send number for factorial to parent.
- Parent process is responsible to calculate the factorial.
- parent process should print the final result

Output

```
  GNU nano 7.2
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

long long factorial (int n){
long long fact = 1;
for (int i = 1; i<= n; i++){
fact *=i;
}
return fact;
}

int main(){
int pipefd[2];
pid_t pid;

if(pipe(pipefd) == -1){
perror ("pipe failed");
return 1;
}
}
pid = fork();
if (pid < 0){
perror ("fork failed");
return 1;
}
if (pid == 0){
close(pipefd[0]);
int num = 5;
write(pipefd[1], &num , sizeof(num));
close(pipefd[1]);
}
else{
close(pipefd[1]);
int received_num;
read(pipefd[0], &received_num, sizeof(received_num));
close(pipefd[0]);

long long result = factorial(received_num);
long long result = factorial(received_num);
printf("factorial of %d is %lld \n", received_num,result);
}
return 0;
}
```

Out put

```
ubuntu@ubuntu:~/Desktop$ gcc lab42.c -o talhabb
ubuntu@ubuntu:~/Desktop$ ./talhabb
factorial of 5 is 120
ubuntu@ubuntu:~/Desktop$
```

task no 04

Code

```
 GNU nano 7.2                                lab42.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
int count_vowels(const char *str){
int count =0;
for (int i=0; str[i] != '\0' ; i++){
char ch = str[i];
if( ch == 'a' || ch == 'e' || ch == 'i' || ch ==  'o' || ch == 'u' ||
 ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch ==  'U' )
{
count++;
}
}
return count ;
}
int main(){
int pipefd[2];
pid_t pid;
if (pipe(pipefd) == -1){
```

```
  if (pipe(pipefd) == -1){
  perror ("pipe failed");
  return 1;
  }
   pid = fork();
  if (pid < 0){
  perror("fork failed ");
  return 1;
  }
  if (pid == 0){
  close(pipefd[0]);
  char str[] = "hello word ";
  write(pipefd[1], str, strlen(str) + 1);
  close(pipefd[1]);
  }else{
  close(pipefd[1]);
  char received_str[100];
  read(pipefd[0], received_str, sizeof(received_str));
  close(pipefd[0]);
```

```
close(pipefd[0]);

int vowel_count = count_vowels(received_str);
printf ("vowel count in  '%s' %d\n" , received_str , vowel_count);
}
return 0;
}


^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Loca
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go
```

Output

```
        |    ^
ubuntu@ubuntu:~/Desktop$ gcc lab42.c -o talhabb
ubuntu@ubuntu:~/Desktop$ ./talhabb
vowel count in  'hello word ' 3
ubuntu@ubuntu:~/Desktop$
```