**OS lab assignment 9**

**Agnim Gupta**

**2028082**

**A-23 CSSE**

**Question 2**

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

#include <string.h>

#include <stdlib.h>

#include <sys/wait.h>

int main(int argc, char \*argv[])

{

int numChildProcesses = 1;

int testArray[4] = {2,7,9,4};

printf("Will use 1 child process; %d pipes.\n", numChildProcesses\*2);

int fd[numChildProcesses\*2][2];

int val = 0, len, i;

for (i=0; i<numChildProcesses\*2; ++i)

{

    if (pipe(fd[i]) < 0)

    {

        perror("Failed to allocate pipes.");

        exit(EXIT\_FAILURE);

    }

}

for (i=0;i<numChildProcesses;i++)

{

    if (fork() == 0)

    {

        int total = 0, xCount = 0;

        while (xCount < 4)

        {

            len = read(fd[i][0], &val, sizeof(val));

            if (len < 0)

            {

                perror("Child: Failed to read data from pipe.\n");

                exit(EXIT\_FAILURE);

            }

            else if (len == 0)

            {

                fprintf(stderr, "Child: Read EOF from pipe\n");

            }

            else

            {

                total += val;

                xCount += 1;

                printf("Child: Recieved %d\tTotal: %d\tCount: %d\n", val, total, xCount);

            }

        }

        printf("Child: Sending %d back\n", total);

        if (write(fd[i][1], &total, sizeof(total)) < 0)

        {

            perror("Child: Failed to write response value");

            exit(EXIT\_FAILURE);

        }

        return EXIT\_SUCCESS;

    }

    if (fork() > 0)

    {

        int total = 0;

        printf("\nParent: Sending numbers to child\n");

        if (write(fd[i][1], &testArray, sizeof(testArray)) != sizeof(testArray))

        {

            perror("Parent: Failed to send value to child ");

            exit(EXIT\_FAILURE);

        }

        len = read(fd[i][0], &val, sizeof(val));

        if (len < 0)

        {

            perror("Parent: failed to read value from pipe");

            exit(EXIT\_FAILURE);

        }

        else if (len == 0)

        {

            fprintf(stderr, "Parent: Read EOF from pipe\n");

        }

        else

        {

            total += val;

            printf("Parent: Received %d\tTotal: %d\n", val, total);

        }

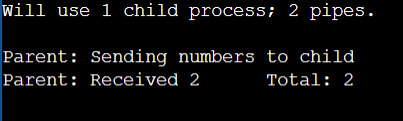
        wait(NULL);

    }

}

}

**Output**

****

**Question 1**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main()

{

    int p1 ;

    int p2 ;

    int p3 ;

    int p4 ;

    int p5 ;

    int p6 ;

    p2=fork();

    if (p2 == 0)

    {

        p4= fork();

                if(p4==0)

                    printf("%d: p1\n", getpid());

        else

        {

            p5=fork();

            if(p5==0)

                printf("%d: p1\n", getpid());

            else

            {

                    printf("%d: p1\n", getpid());

            }

        }

    }

    else

    {

        p3=fork();

        if(p3==0){

            p6=fork();

            if(p6==0)

                printf("%d: p1\n", getpid());

            else

                printf("%d: p1\n", getpid());

        }

        else

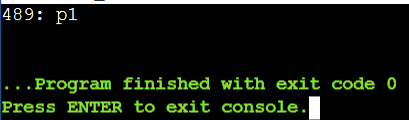
            printf("%d: p1\n", getpid());

    }

    return 0;

}

**Output**

****