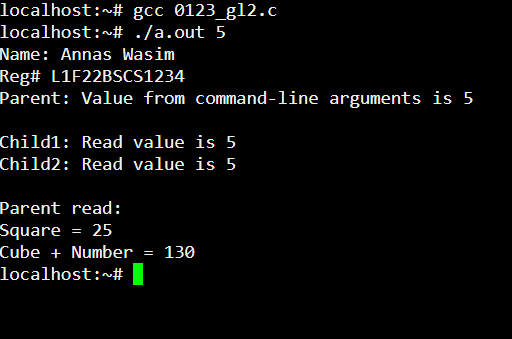
Create a C program that utilizes command line arguments to receive an integer from the user (parent process’s responsibility). Parent process forks two child processes, implement a solution where the parent process sends this integer to both children through separate pipes. Child 1 (c1) computes the square of the number and sends it back to the parent through one pipe, while Child 2 (c2) calculates the cube of the number added to itself and communicates this result to the parent through a different pipe. Display both results received from the children in the parent process. [CLO-1]

(Marks: 5+5+5 = 15)

***NOTE: Do not cheat. Submit only .c file(yourRegNo\_gl2.c). Time allowed is 35 mins (including submission).***

**Sample Output:**



**Solution:**

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include<unistd.h>  #include<sys/types.h>  #include<sys/wait.h>  int main(int argc, char\* argv[]){  printf("Name: Annas Wasim\nReg# L1F22BSCS1234\n");  pid\_t pid;  int fd[2],fd1[2],fd2[2],fd3[2];  pipe(fd);pipe(fd1);pipe(fd2);pipe(fd3);  pid = fork();  if(pid == 0){  close(fd[1]);  int a;  int sq;  read(fd[0],&a,sizeof(a));  printf("Child1:\tRead value is %d\n",a);  sq = a\*a;  //child1 write  close(fd2[0]);  write(fd2[1],&sq,sizeof(sq));  }  else{  pid\_t pid1 = fork();  if(pid1 == 0){  close(fd1[1]);  int b;  int cb;  read(fd1[0],&b,sizeof(b));  printf("Child2:\tRead value is %d\n",b);  cb = (b\*b\*b)+b;  //child2 write  close(fd3[0]);  write(fd3[1],&cb,sizeof(cb));  }  else{  //write op to child1 and child2  int c1,c2;  int value = atoi(argv[1]);  printf("Parent:\tValue from command-line arguments is %d\n\n",value);  close(fd[0]);  close(fd1[0]);  write(fd[1],&value,sizeof(value));  write(fd1[1],&value,sizeof(value));  wait(NULL);  //parent read  close(fd2[1]);  close(fd3[1]);  read(fd2[0],&c1,sizeof(c1));  read(fd3[0],&c2,sizeof(c2));  printf("\nParent read:\nSquare = %d\nCube + Number = %d\n",c1,c2);  }  }  return 0;  } |