

3

Student Name: Ameer Nasrallah

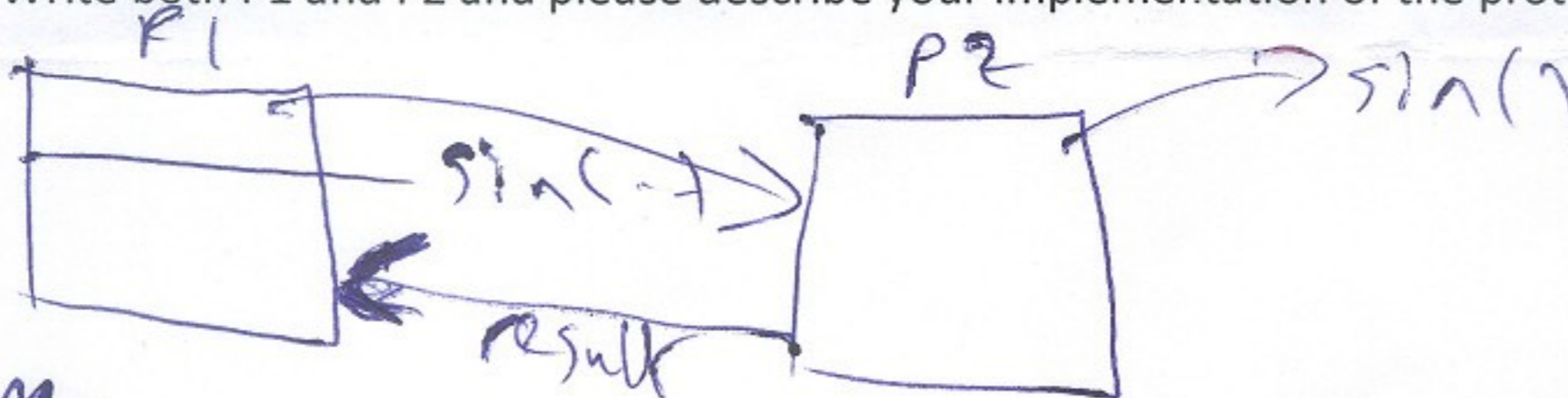
Student number: 1090338

4/3/2013

Process P1 is trying to hide some of its system calls from debugging. In order to do that process P2 is created. P1 sends its request to run a system call and the parameters to P2 via IPC, P2 performs the operation and returns the results to P1. For this case assume the system calls P1 needs from P2 are

```
double sin(double);
double cos(double);
int abs(int);
(int)(float) // converting a float to int
```

Write both P1 and P2 and please describe your implementation of the protocol between P1 and P2



Using FIFO, P1 will write to a file some data, and then P2 will read this data and based on it will return the result of some required function.

```
P1:
int r = mknod("/tmp/MYFIFO", S_IFIFO|0666, 0);
if (r == 0) {
    FILE *out = fopen("myfile.txt", "w");
    fwrite(out, "sin 50\n cos 45\n abs -7\n (int) 8.7");
}
```

```
P2: char *data;
if (mknod("/tmp/MYFIFO", S_IFIFO|0666, 0) == 0) {
    FILE *in = fopen("myfile.txt", "r");
    while (read(in, data)) {
        if (data == "sin") return sin(read(in, data));
        else if (data == "cos") return cos(read(in, data));
    }
}
```



else if (data == "abs")

return (abs(read(in, data)));

else if (data == "int")

return (atoi(read(in, data)));

Student Name: Aymer Nasserwan Student number: 1090338

Process P1 is trying to hide some of its system calls from debugging. In order to do this, process P2 is created. P1 sends its request to run a system call and the parameters to P2 via the pipe. P2 performs the operation and returns the results to P1. For this case assume the system call is read from P2 side.

double sin(double);  
double cos(double);  
int atoi(int);

(int)float // converting a float to int

Write both P1 and P2 and please describe your implementation of the protocol between P1 and P2.



Process P1 will write to a pipe  
Send data, and then  
P2 will read the data  
and based on it will  
return the result  
of some system call  
P1 will read the result  
and based on it will  
write back to a pipe

P1:  
int main() {  
 int fd[2];  
 pipe(fd);  
 // ...  
}

P2: char \*data;  
if (read(fd[0], data, 1024) > 0) {  
 if (data == "sin") {  
 double x = atof(data);  
 double y = sin(x);  
 write(fd[1], y, sizeof(y));  
 }  
 else if (data == "cos") {  
 double x = atof(data);  
 double y = cos(x);  
 write(fd[1], y, sizeof(y));  
 }  
 else if (data == "atoi") {  
 int x = atoi(data);  
 write(fd[1], x, sizeof(x));  
 }  
}