



BÁO CÁO THỰC HÀNH

Buổi 5

Môn:	Nhập môn Hệ điều hành
Nhóm:	N3T01
Họ và tên:	Lưu Hữu Trí
MSSV:	52200167
Số bài hoàn thành:	3/6 (50%)

Câu 1:

```
bai1.c x bai2.c x bai3.c x
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <sys/wait.h>
6
7 int main(int argc, char* argv[])
8 {
9     int fp1[2], fp2[2];
10    int pid, i, a;
11
12    if (pipe(fp1) == 0 & pipe(fp2) == 0) {
13        pid = fork();
14        if (pid < 0) {
15            printf("Fork failed\n");
16            return -1;
17        }
18        else if (pid == 0) {
19            close(fp1[1]);
20            read(fp1[0], &a, sizeof(a));
21            char buffer[a + 1];
22            close(fp1[0]);
23
24            close(fp2[1]);
25            read(fp2[0], &buffer, sizeof(buffer));
26            buffer[a] = '\0';
27            printf("Read from parents: %s\n", buffer);
28            close(fp2[0]);
29        }
30        else {
31            char tem[256];
32            int a;
33            for (i = 1; i < argc; i++) {
34                strcat(tem, argv[i]);
35                strcat(tem, " ");
36            }
37
38            a = strlen(tem);
39            close(fp1[0]);
40            write(fp1[1], &a, sizeof(a));
41            close(fp1[1]);
42
43            close(fp2[0]);
44            write(fp2[1], tem, a);
45            close(fp2[1]);
46            wait(NULL);
47        }
48    }
49    return 0;
50 }
```

C ▾ Tab Width: 4 ▾ Ln 50, Col 2 INS

```
root@ubuntu: /home/lab5
root@ubuntu:/home/lab5# gcc -c bai1.c
root@ubuntu:/home/lab5# gcc -o bai1.out bai1.o
root@ubuntu:/home/lab5# ./bai1.out Luu Huu Tri
Read from parents: 64L_~|Luu Huu Tri
root@ubuntu:/home/lab5#
```

Câu 2:

```
bai1.c x bai2.c x bai3.c x
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/wait.h>
4 #include <unistd.h>
5
6 int factorial(int n) {
7     if (n == 0 || n == 1) return 1;
8     if (n == 2) return 2;
9     return n * factorial(n - 1);
10 }
11
12 int main(int argc, char ** argv)
13 {
14     int fp1[2], fp2[2], pid;
15
16     if (pipe(fp1) == 0 && pipe(fp2) == 0) {
17         pid = fork();
18         if (pid < 0) {
19             printf("Fork failed\n");
20             return -1;
21         }
22         else if (pid == 0) {
23             int a;
24             close(fp1[1]);
25             read(fp1[0], &a, sizeof(a));
26             printf("Data received from children is: %d\n", a);
27             int n = factorial(a);
28
29             close(fp1[0]);
30
31             close(fp2[0]);
32             write(fp2[1], &n, sizeof(n));
33             close(fp2[1]);
34         }
35         else {
36             close(fp1[0]);
37             int a = atoi(argv[1]);
38             write(fp1[1], &a, sizeof(a));
39             close(fp1[1]);
```

```

39
40         close(fp2[1]);
41         int t;
42         read(fp2[0], &t, sizeof(t));
43         printf("%d! = %d\n", a, t);
44         close(fp2[0]);
45     }
46 }
47
48 return 0;
49 }

```

Bracket match found on line: 13

C ▾

Tab Width: 4 ▾

Ln 49, Col 2

INS

```

root@ubuntu: /home/lab5
root@ubuntu:/home/lab5# gcc -c bai2.c
root@ubuntu:/home/lab5# gcc -o bai2.out bai2.o
root@ubuntu:/home/lab5# ./bai2.out 5
Data received from children is: 5
5! = 120
root@ubuntu:/home/lab5#

```

Câu 3:

```

bai1.c x bai2.c x bai3.c x
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <string.h>
5
6 int main(int argc, char ** argv)
7 {
8     int fp1[2], fp2[2], pid;
9
10    if (pipe(fp1) == 0 && pipe(fp2) == 0) {
11        pid = fork();
12        if(pid == 0) {
13            close(fp1[1]);
14            int a, b;
15            char temp[2];
16            read(fp1[0], &a, sizeof(a));
17            read(fp1[0], &b, sizeof(b));
18            read(fp1[0], &temp, sizeof(temp));
19            close(fp1[0]);

```

```

20
21     int c = 0;
22     if (strcmp(temp, "+") == 0) {
23         c = a + b;
24     }
25     else if (strcmp(temp, "-") == 0) {
26         c = a - b;
27     }
28     else if (strcmp(temp, "*") == 0) {
29         c = a * b;
30     }
31     else if (strcmp(temp, "/") == 0) {
32         c = a / b;
33     }
34     close(fp2[0]);
35     write(fp2[1], &c, sizeof(c));
36     close(fp2[1]);
37 }
38 else {
39     close(fp1[0]);
40     int a = atoi(argv[1]);
41     int b = atoi(argv[2]);
42     write(fp1[1], &a, sizeof(a));
43     write(fp1[1], &b, sizeof(b));
44     write(fp1[1], argv[3], sizeof(argv[3]));
45     close(fp1[1]);
46
47     int n;
48     close(fp2[1]);
49     read(fp2[0], &n, sizeof(n));
50     close(fp2[0]);
51     printf("%d %s %d = %d\n", a, argv[3], b, n);
52 }
53 }
54 return 0;
55 }

```

C ▾ Tab Width: 4 ▾

Ln 46, Col 9

INS

```

root@ubuntu: /home/lab5
root@ubuntu:/home/lab5# gcc -c bai3.c
root@ubuntu:/home/lab5# gcc -o bai3.out bai3.o
root@ubuntu:/home/lab5# ./bai3.out
Segmentation fault (core dumped)
root@ubuntu:/home/lab5# ./bai3.out 21 9 +
21 + 9 = 30
root@ubuntu:/home/lab5# |

```