

Họ và tên: Nguyễn Văn Hưng

Mã số sinh viên: 23520569

Lớp: ATTN2023

HỆ ĐIỀU HÀNH BÁO CÁO LAB 3

CHECKLIST (Đánh dấu x khi hoàn thành)

Lưu ý mỗi câu phải làm đủ 3 yêu cầu

I. CLASSWORK

| | BT 1 | BT 2 | BT 3 | BT 4 |
|----------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Trình bày cách làm | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Chụp hình minh chứng | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Giải thích kết quả | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

II. HOMEWORK

| | |
|----------------------|--------------------------|
| | 3.6 |
| Trình bày cách làm | <input type="checkbox"/> |
| Chụp hình minh chứng | <input type="checkbox"/> |
| Giải thích kết quả | <input type="checkbox"/> |

Tự chấm điểm: 9

**Lưu ý: Xuất báo cáo theo định dạng PDF, đặt tên theo cú pháp:*

<MSSV>_LABx.pdf

I. CLASSWORK

1.

- Ví dụ 3-1:
 - Code: [test_fork.c](#)
 - Kết quả:

```
→ Lab3 ls
23520569_Lab3.docx 'Lab 3 v2023.pdf' test_fork test_fork.c
→ Lab3 code test_fork.c
→ Lab3 g++ -o test_fork test_fork.c
→ Lab3 ./test_fork 1 2 3
PARENTS | PID = 756 | PPID = 304
PARENTS | There are 3 arguments
CHILDREN | PID = 757 | PPID = 756
CHILDREN | List of arguments:
1
2
3
```

- Giải thích code:
 - + Hàm fork() thực hiện tạo ra tiến trình mới là tiến trình con.
 - + pid > 0: Đây là tiến trình cha, pid chứa giá trị Process ID của tiến trình con
 - Thực hiện in ra Process ID và Parent Process ID của tiến trình cha
 - In ra số lượng tham số được truyền vào.
 - + pid == 0: Đây là tiến trình con, pid bằng 0 trong tiến trình con.
 - Thực hiện in ra Process ID và Parent ID của tiến trình con
 - In ra giá trị của các tham số được truyền vào.

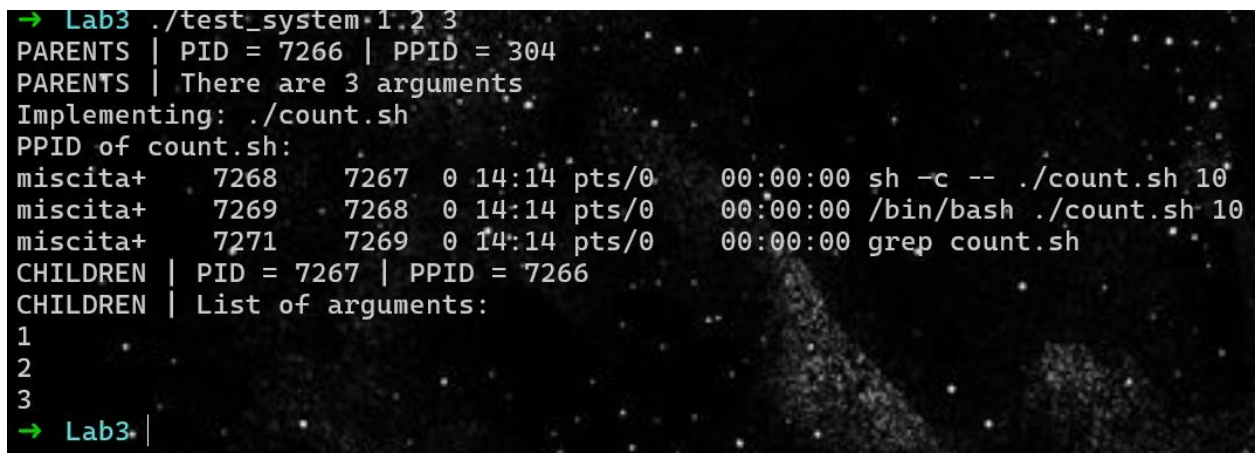
- Ví dụ 3-2:
 - Code: [test_execl.c](#)
 - Kết quả:

```
→ Lab3 g++ -o test_execl test_execl.c
→ Lab3 ./test_execl 1 2 3
PARENTS | PID = 4510 | PPID = 304
PARENTS | There are 3 arguments
Implementing: ./count.sh
PPID of count.sh:
miscita+ 4511 4510 0 14:00 pts/0 00:00:00 /bin/bash ./count.sh 10
miscita+ 4513 4511 0 14:00 pts/0 00:00:00 grep count.sh
→ Lab3 |
```

- Giải thích code:
 - + fork() tạo ra tiến trình con.
 - + pid > 0: Tiến trình cha in ra Process ID và Parent Process ID
In ra số lượng tham số được truyền vào
wait(NULL) đợi đến khi tiến trình con hoàn thành rồi tiếp tục chạy.
 - + pid == 0: Gọi hàm execl thay thế chương trình hiện tại bằng count.sh
count.sh thực hiện ps -ef để in ra các tiến trình đang chạy và lọc
bằng grep count.sh
In từ 1 đến (\$1 = 10) vào file count.txt

- Ví dụ 3-3:

- Code: [test_system.c](#)
- Kết quả:



```
→ Lab3 ./test_system 1 2 3
PARENTS | PID = 7266 | PPID = 304
PARENTS | There are 3 arguments
Implementing: ./count.sh
PPID of count.sh:
miscita+ 7268 7267 0 14:14 pts/0 00:00:00 sh -c -- ./count.sh 10
miscita+ 7269 7268 0 14:14 pts/0 00:00:00 /bin/bash ./count.sh 10
miscita+ 7271 7269 0 14:14 pts/0 00:00:00 grep count.sh
CHILDREN | PID = 7267 | PPID = 7266
CHILDREN | List of arguments:
1
2
3
→ Lab3 |
```

- Giải thích:
 - + fork() tạo ra các tiến trình như các ví dụ trên.
 - + Tiến trình cha in ra PID, PPID và số lượng tham số truyền vào.
 - + system("./count.sh 10") tạo ra 1 tiến trình mới nên khi tiến trình mới hoàn thành xong thì chương trình con vẫn tiếp tục chạy.

- Ví dụ 3-4:
 - Code: [test_fork_wait.c](#)
 - Kết quả:

```
→ Lab3 g++ -o test_fork_wait test_fork_wait.c
→ Lab3 ./test_fork_wait
Child process, pid=0
Parent Proces, pid=8284
→ Lab3 ./test_fork_wait
Child process, pid=0
Parent Proces, pid=8303
```

- Giải thích:
 - + fork() tạo ra tiến trình con
 - + pid = 0, tiến trình con in ra pid của nó (luôn bằng 0)
 - + pid > 0, tiến trình cha thực hiện lệnh wait(NULL) đợi khi tiến trình con hoàn thành mới chạy tiếp, sau đó in ra pid của tiến trình cha.

2.

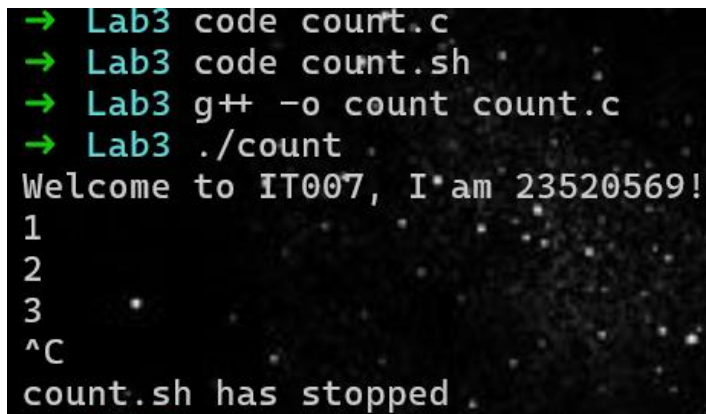
- Code: [time.c](#)
- Kết quả:

```
→ Lab3 ./time ls
23520569_Lab3.docx count.sh test_execl test_fork test_fork_wait test_system time '~\WRL0005.tmp'
'Lab 3 v2023.pdf' count.txt test_execl.c test_fork.c test_fork_wait.c test_system.c time.c
Thời gian thực thi: 0.002982 giây
→ Lab3
```

- Giải thích:
 - + gettimeofday(&start, NULL) tính thời gian bắt đầu
 - + fork() tạo ra tiến trình con.
 - + Chương trình con thực hiện <command> được nhập bằng execlp
 - + Chương trình cha gọi wait(NULL) đợi đến khi tiến trình con hoàn thành.
 - + gettimeofday(&end, NULL); tính thời gian kết thúc
 - + Sau đó lấy thời gian kết thúc trừ thời gian bắt đầu là ra được kết quả.

3.

- Code: [count.c](#)
- Kết quả:



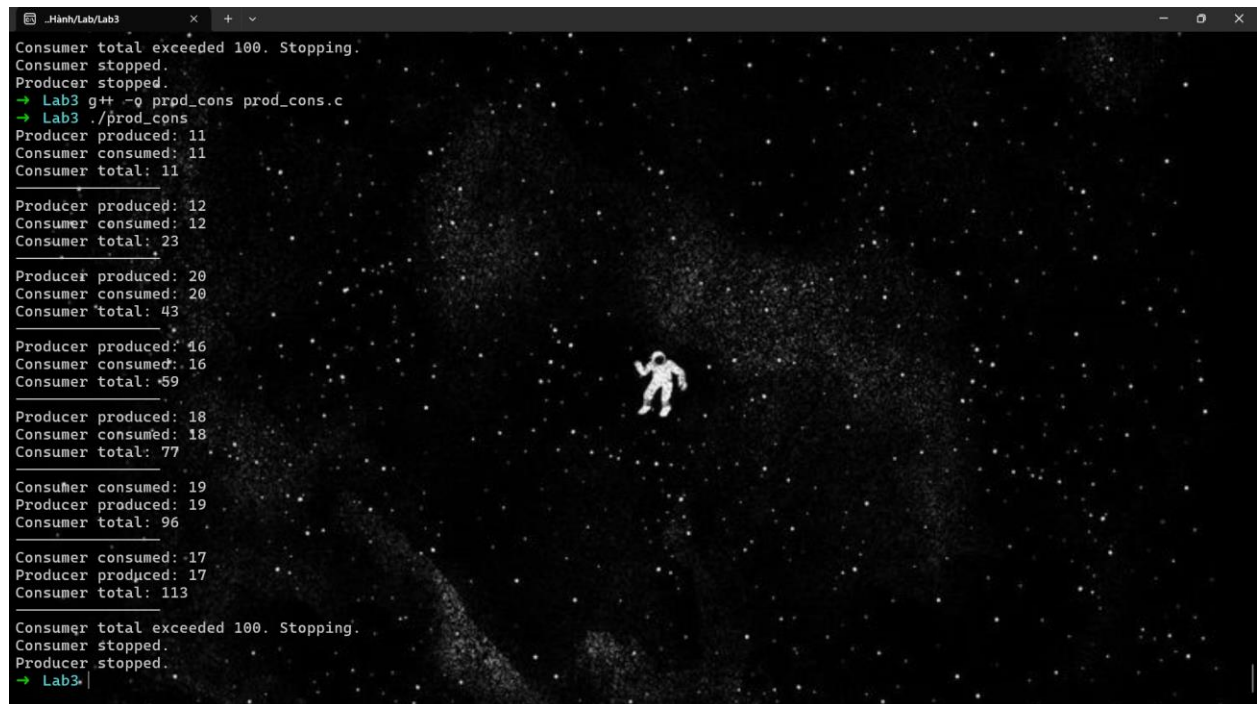
```
→ Lab3 code count.c
→ Lab3 code count.sh
→ Lab3 g++ -o count count.c
→ Lab3 ./count
Welcome to IT007, I am 23520569!
1
2
3
^C
count.sh has stopped
```

- Giải thích:
 - + fork() tạo ra chương trình con.
 - + Chương trình con thực `execl("./count.sh", "./count.sh", "120", NULL);`
 - + Khi nhấn Ctrl+C SIGINT được gửi đến tiến trình cha và tiến trình con, tiến trình con bị dừng ngay lập tức nhưng tiến trình cha vẫn chạy.
 - + `signal(SIGINT, handle_sigint)` tiến trình cha khi nhận được SIGINT sẽ gọi hàm `handle_sigint`.
 - + `handle_sigint` thực hiện `kill(child_pid, SIGKILL)` sẽ gửi tín hiệu SIGKILL đến `count.sh` và buộc nó phải dừng
 - + Sau đó in ra `count.sh has stopped`

4.

- Code: [product-consumer](#)

- Kết quả:



```
Consumer total exceeded 100. Stopping.
Consumer stopped.
Producer stopped.
→ Lab3 g++ -o prod_cons prod_cons.c
→ Lab3 ./prod_cons
Producer produced: 11
Consumer consumed: 11
Consumer total: 11

Producer produced: 12
Consumer consumed: 12
Consumer total: 23

Producer produced: 20
Consumer consumed: 20
Consumer total: 43

Producer produced: 16
Consumer consumed: 16
Consumer total: 59

Producer produced: 18
Consumer consumed: 18
Consumer total: 77

Consumer consumed: 19
Producer produced: 19
Consumer total: 96

Consumer consumed: 17
Producer produced: 17
Consumer total: 113

Consumer total exceeded 100. Stopping.
Consumer stopped.
Producer stopped.
→ Lab3
```

- Giải thích:

- + Tạo ra `shared_memory` 10 bytes để chia sẻ dữ liệu giữa tiến trình cha và tiến trình con.
- + `fork()` tạo ra tiến trình con.
- + Chương trình cha thực hiện random một số ngẫu nhiên trong khoảng từ 10 đến 20, đợi đến khi `*shared_memory` bằng 0 hoặc -1 thì tiếp tục chạy.
- + `*shared_memory` bằng 0 thì tiến trình con đã nhận dữ liệu và tính tổng.
- + `*shared_memory` bằng -1 thì tiến trình con đã kết thúc.
- + Tiến trình con thực hiện đợi cho đến khi `*shared_memory` khác 0 có nghĩa là tiến trình cha để cho dữ liệu vào.
- + Sau đó, tiến trình con thực hiện tính tổng
- + Nếu tổng lớn hơn 100 thì gán `*shared_memory` bằng -1 và `break` để kết thúc chương trình. Ngược lại, gán `*shared_memory` bằng 0 thể hiện đã xử lý dữ liệu.

....

II. HOMEWORK

1.

- Code: [collatz.c](#)
- Kết quả:

```
→ Lab3 g++ -o collatz collatz.c
→ Lab3 ./collatz 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
→ Lab3 ./collatz 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
→ Lab3 ./collatz 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
→ Lab3 ./collatz 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
→ Lab3 ./collatz 35
35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1
→ Lab3 ./collatz 8
8, 4, 2, 1
→ Lab3 ./collatz -3
Number must be greater than 0
→ Lab3 ./collatz
Usage: ./collatz <number>
→ Lab3 |
```

- Giải thích:
 - + Kiểm tra nếu argc < 2 thì yêu cầu nhập tham số.
 - + Kiểm tra nếu input < 1 thì yêu cầu tham số số lớn hơn 0
 - + Tạo ra shared_memory và fork() để tạo tiến trình con.
 - + Tiến trình cha đợi *(shared_memory + 1) bằng 1 mới xử lý, tiến trình con thì bằng 0
 - + Tiến trình cha nhận giá trị hiện tại *(shared_memory), in ra màn hình sau đó kiểm tra giá trị hiện tại bằng 1 thì không xử lý nữa.
 - + Tiến trình cha đợi tiến trình con kết thúc mới dừng chương trình.
 - + Tiến trình con đợi *(shared_memory + 1) bằng 0 mới xử lý.
 - + Tiến trình con nhận giá trị tại *(shared_memory) tính toán giá trị tiếp theo và thay đổi *(shared_memory + 1) bằng 1.
 - + Nếu trước khi xử lý bằng 1 hoặc bằng 2 thì dừng chương trình.