

### Hướng dẫn Lab 3.1 – Quản lý tiến trình

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Khái niệm tiến trình	Ch3: Process Creation	Sử dụng image Ubuntu 16/18 Sử dụng Windows 10
Quan sát tiến trình trong hệ thống.	Ch3: Process Creation	

### Cập nhật

2021.Jun.2	Thêm phần theo dõi tiến trình.
------------	--------------------------------

**Yêu cầu sinh viên:** Hiểu lý thuyết liên quan. Biết biên dịch các đoạn code mẫu và thực thi. Biết cách quan sát các tiến trình bên trong hệ thống Windows và Ubuntu. Áp dụng các lệnh được giới thiệu để hoàn thành bài tập.

**Đánh giá sinh viên:** Trả lời các vấn đề lý thuyết. Kỹ năng thực hành. Bài tập.

**Yêu cầu nộp bài:** các tập tin mã nguồn .c và tập tin khả thi .out của các “ví dụ” và bài tập cuối hướng dẫn. Ảnh chụp màn hình các thông tin tiến trình.

### Preferences

[1] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2018], Operating System Concepts, 10th edition, John Wiley & Sons, New Jersey.

*Programming Problems of Chapter 3.*

[2] Greg Gagne , [2019], GitHub OS-BOOK OSC10e, Westminster College, United States

Access <https://github.com/greggagne/osc10e> in September 2019.

[3] Linux manual page, [2021], on getpid(2)

Access <https://man7.org/linux/man-pages/man2/getpid.2.html>

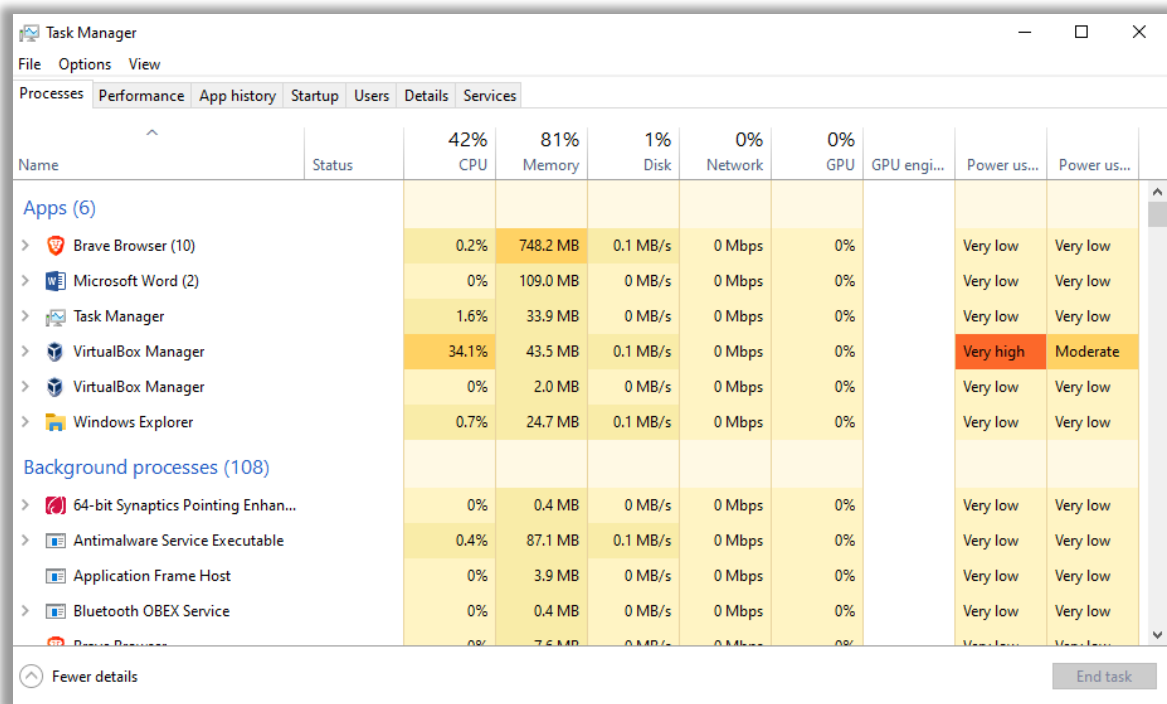
## 1. Quản lý tiến trình trong GUI.

?	- Tiến trình là gì?	Ch3 p.106	Slide Ch3.4
	- Tiến trình sinh ra khi nào? Và kết thúc khi nào?	Ch3 p.116	Slide Ch3.20
	- Sự khác nhau của Tiến trình và Chương trình là gì?	Ch3 p.121	Slide Ch3.25

**Yêu cầu 1:** Mở ứng dụng Task Manager của Windows bằng cách Nhấn tổ hợp Ctrl + Alt + Del và chọn Task Manager.

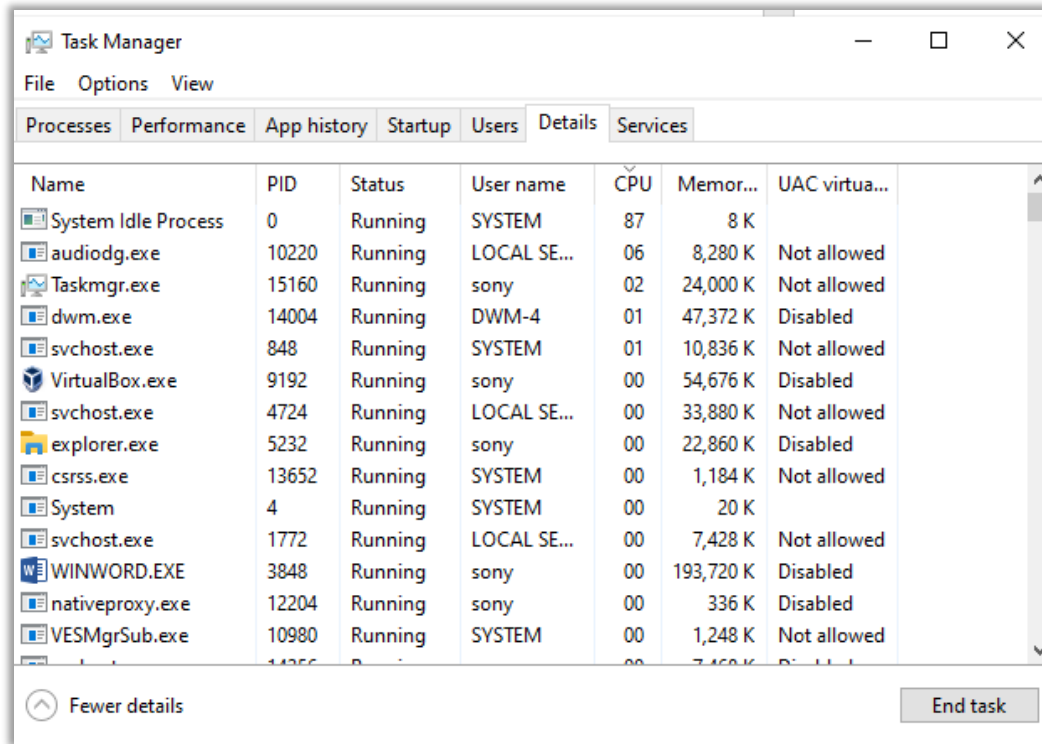
### Quan sát tab Process:

- Giải thích Apps và Background Process khác nhau như thế nào?
- Ứng dụng nào đang sử dụng CPU nhiều nhất, ứng dụng nào đang được cấp RAM nhiều nhất.
- Thao tác trên một vài ứng dụng (chẳng hạn: trình duyệt) và quan sát thông tin thay đổi trong thẻ Process.
- Sắp xếp các ứng dụng theo Mức độ sử dụng CPU từ cao xuống thấp.



Hình 1. Thẻ Process của ứng dụng Task Manager.

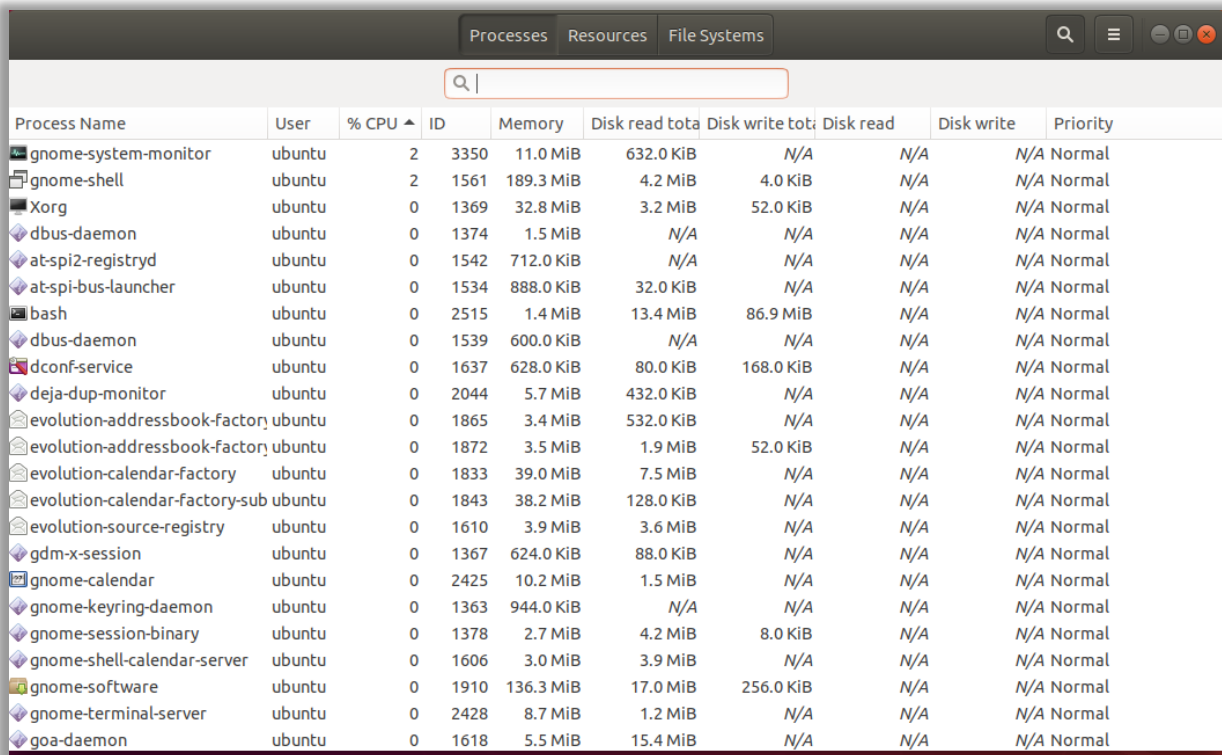
## Quan sát tab Details



Hình 2. Thẻ Details của ứng dụng Task Manager.

- Tiến trình A (hay một tiến trình nào đó), có ID là bao nhiêu, ai là người khởi chạy và trạng thái của nó là gì?
- Tiến trình “System Idle Process” là một tiến trình đặc biệt, luôn sử dụng “phần còn lại” của CPU. Tại sao cần có một tiến trình như vậy bên trong Hệ điều hành?

**Yêu cầu 2:** Mở ứng dụng System Monitor của Ubuntu.



Process Name	User	% CPU	ID	Memory	Disk read tota	Disk write tot	Disk read	Disk write	Priority
gnome-system-monitor	ubuntu	2	3350	11.0 MiB	632.0 KiB	N/A	N/A	N/A	Normal
gnome-shell	ubuntu	2	1561	189.3 MiB	4.2 MiB	4.0 KiB	N/A	N/A	Normal
Xorg	ubuntu	0	1369	32.8 MiB	3.2 MiB	52.0 KiB	N/A	N/A	Normal
dbus-daemon	ubuntu	0	1374	1.5 MiB	N/A	N/A	N/A	N/A	Normal
at-spi2-registryd	ubuntu	0	1542	712.0 KiB	N/A	N/A	N/A	N/A	Normal
at-spi-bus-launcher	ubuntu	0	1534	888.0 KiB	32.0 KiB	N/A	N/A	N/A	Normal
bash	ubuntu	0	2515	1.4 MiB	13.4 MiB	86.9 MiB	N/A	N/A	Normal
dbus-daemon	ubuntu	0	1539	600.0 KiB	N/A	N/A	N/A	N/A	Normal
dconf-service	ubuntu	0	1637	628.0 KiB	80.0 KiB	168.0 KiB	N/A	N/A	Normal
deja-dup-monitor	ubuntu	0	2044	5.7 MiB	432.0 KiB	N/A	N/A	N/A	Normal
evolution-addressbook-factory	ubuntu	0	1865	3.4 MiB	532.0 KiB	N/A	N/A	N/A	Normal
evolution-addressbook-factory	ubuntu	0	1872	3.5 MiB	1.9 MiB	52.0 KiB	N/A	N/A	Normal
evolution-calendar-factory	ubuntu	0	1833	39.0 MiB	7.5 MiB	N/A	N/A	N/A	Normal
evolution-calendar-factory-sub	ubuntu	0	1843	38.2 MiB	128.0 KiB	N/A	N/A	N/A	Normal
evolution-source-registry	ubuntu	0	1610	3.9 MiB	3.6 MiB	N/A	N/A	N/A	Normal
gdm-x-session	ubuntu	0	1367	624.0 KiB	88.0 KiB	N/A	N/A	N/A	Normal
gnome-calendar	ubuntu	0	2425	10.2 MiB	1.5 MiB	N/A	N/A	N/A	Normal
gnome-keyring-daemon	ubuntu	0	1363	944.0 KiB	N/A	N/A	N/A	N/A	Normal
gnome-session-binary	ubuntu	0	1378	2.7 MiB	4.2 MiB	8.0 KiB	N/A	N/A	Normal
gnome-shell-calendar-server	ubuntu	0	1606	3.0 MiB	3.9 MiB	N/A	N/A	N/A	Normal
gnome-software	ubuntu	0	1910	136.3 MiB	17.0 MiB	256.0 KiB	N/A	N/A	Normal
gnome-terminal-server	ubuntu	0	2428	8.7 MiB	1.2 MiB	N/A	N/A	N/A	Normal
goa-daemon	ubuntu	0	1618	5.5 MiB	15.4 MiB	N/A	N/A	N/A	Normal

Hình 3. Ứng dụng *gnome-system-monitor*.

- Thực hiện các thao tác như ở Yêu cầu 1.
- Có những điểm giống và điểm khác nào giữa Task Manager và System Monitor?

## 2. Quản lý tiến trình trong CLI

**Yêu cầu 3:** Thực thi các lệnh quản lý tiến trình trong Terminal của Ubuntu.

- Liệt kê tiến trình:

```
> ps [options]
```

- Các option:

- -a: hiển thị các tiến trình của user được liên kết
- -e: hiển thị tất cả tiến trình
- -f: hiển thị PID của tiến trình cha và thời điểm bắt đầu
- -l: tương tự -f

- Hủy tiến trình:

```
> kill [-signal | -s signal] pid
```

Thường kết hợp với lệnh ps để lấy ID tiến trình

- Các signal:

- 2: tương đương CTRL + C
- 9: buộc kết thúc
- 15: mặc định – kết thúc êm ái
- 19: tạm dừng
- Kill – l: liệt kê tất cả các signal

- Ví dụ:

```
kill -9 11234 (kill tiến trình có pid = 11234)
```

```
kill -s 9 11234
```

### 3. Những lời gọi Quản lý tiến trình

**Yêu cầu 4:** Sử dụng các hàm lấy thông tin tiến trình. Thư viện “unistd.h” chứa các lời gọi hệ thống tạo tiến trình fork() và xem ID của chúng getpid() và xem ID của tiến trình cha getppid().

**Hướng dẫn:** Tải về tập tin *LAB\_IntroOS/LAB\_3/task1\_4.c*, sau đó biên dịch và thực thi để biết ID của tiến trình này cùng với ID của tiến trình cha.

- Dòng 5 là thư viện chứa lời gọi ở dòng 7, 8.
- Dòng 7 chứa lời gọi trả về ID của tiến trình đang chạy.
- Dòng 8 chứa lời gọi trả về ID của tiến trình cha.

```
5     #include <unistd.h>
7     printf("Current process ID: %d\n", getpid());
8     printf("Parent process ID: %d\n", getppid())
```

?	<ul style="list-style-type: none"><li>- ID của tiến trình là bao nhiêu?</li><li>- ID của tiến trình cha là bao nhiêu?</li><li>- Mở System Monitor và tìm xem các tiến trình nào có ID đã được nêu trên.</li></ul>	Ch3 p.107	Slide Ch3.10
---	---	-----------	--------------

**Yêu cầu 5:** thêm vào lệnh sleep(5) sau dòng 7 và trước dòng 8. Biên dịch và chạy lại chương trình, cùng lúc chuyển sang tiện ích “System Monitor” tìm kiếm các thông tin ID của tiến trình, quan sát sự thay đổi khi tiến trình kết thúc sau 5 giây.

Ghi chú: tham số 5 trong lệnh sleep nghĩa là chương trình tạm dừng 5 giây, SV có thể tăng lên nếu không thao tác kịp hoặc giảm xuống nếu muốn thí nghiệm diễn ra nhanh hơn.

## Bài tập lập trình

1. Thêm vào lệnh while(1) vào cuối chương trình trong yêu cầu 4 để tiến trình chạy mãi mãi. Vào System Monitor và thao tác để kết thúc tiến trình này.
2. Tiếp tục thực thi chương trình có vòng lặp while(1); đưa tiến trình này vào “background” thông qua việc gửi tín hiệu SIGTSTP bằng cách nhấn Ctrl + Z. Sử dụng lệnh ps để xác định PID của nó và sử dụng lệnh kill để kết thúc nó.
3. Xem lại tiến trình phía trên, khi thực thi hệ thống cấp bộ nhớ dung lượng bao nhiêu bytes? Hãy chỉnh sửa chương trình, khai báo mảng 1 triệu số nguyên và khởi tạo giá trị cho chúng. Khi thực thi chương trình chỉnh sửa, hệ thống cấp bộ nhớ dung lượng bao nhiêu bytes? Hãy lý giải về sự thay đổi này.

### CHECKLIST LAB

- ☐ Khởi động Ubuntu.
- ☐ Chạy Terminal.
- ☐ Biên tập một tập tin văn bản .txt, .c
- ☐ Biên dịch và Liên kết một mã nguồn .c
- ☐ Tạo thư viện liên kết tĩnh.
- ☐ Tạo thư viện liên kết động.
- ☐ Chạy một chương trình đã biên dịch.
- ☐ Truyền đối số qua dòng lệnh.
- ☐ Tạo tiến trình con.
- ☐ Cấu trúc mã nguồn cho tiến trình cha và tiến trình con.
- ☐ Tạo cây tiến trình
- ☐ Đồng bộ kết thúc tiến trình cha/con.
- ☐ Gọi chạy một lệnh từ trong chương trình.