[**1. Xem danh sách các process đang chạy trên hệ thống bằng ps 1**](#_utrkipgz0hh6)

[**2. Xem danh sách các process đang chạy trên hệ thống bằng lệnh pstree 2**](#_8mgtffwbd9is)

[**3. Xem cách danh sách process đang chạy trên hệ thống bằng lệnh top 3**](#_bpkr7gkfebwr)

[**5. Xem tỉ lệ CPU, RAM hệ thống đang sử dụng của từng process đang chạy: 6**](#_uzt73abyykn3)

[**top 6**](#_3uuobw9lflkd)

[**6. In thông tin process đang sử dụng nhiều CPU nhất 7**](#_33pu9imckff6)

[**7. Đếm số process đang thực thi trên máy 7**](#_3zkz9xjktn84)

[**8. Đếm số process của user root đang thực thi trên máy 7**](#_mfbgumytxwza)

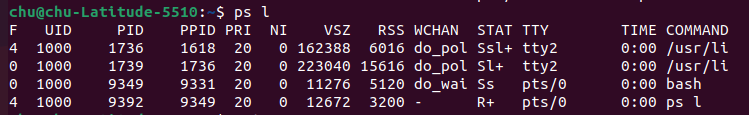
[**9. Đếm số process “httpd” của user root đang thực thi trên máy 7**](#_txtinn6hn2zl)

[**10. Cho biết có bao nhiêu process đang chạy trong hệ thống 7**](#_qm6zqwez3zue)

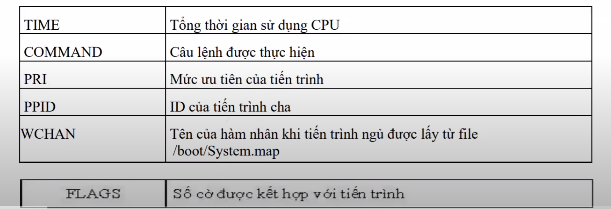
[**tạo file 7**](#_8m8qsnyh6rm5)

[**11. Thực thi lệnh in thông tin tất cả các tập tin, thư mục trong hệ thống. Kết quả lưu vào file 8**](#_4703qebudxec)

# Xem danh sách các process đang chạy trên hệ thống bằng ps

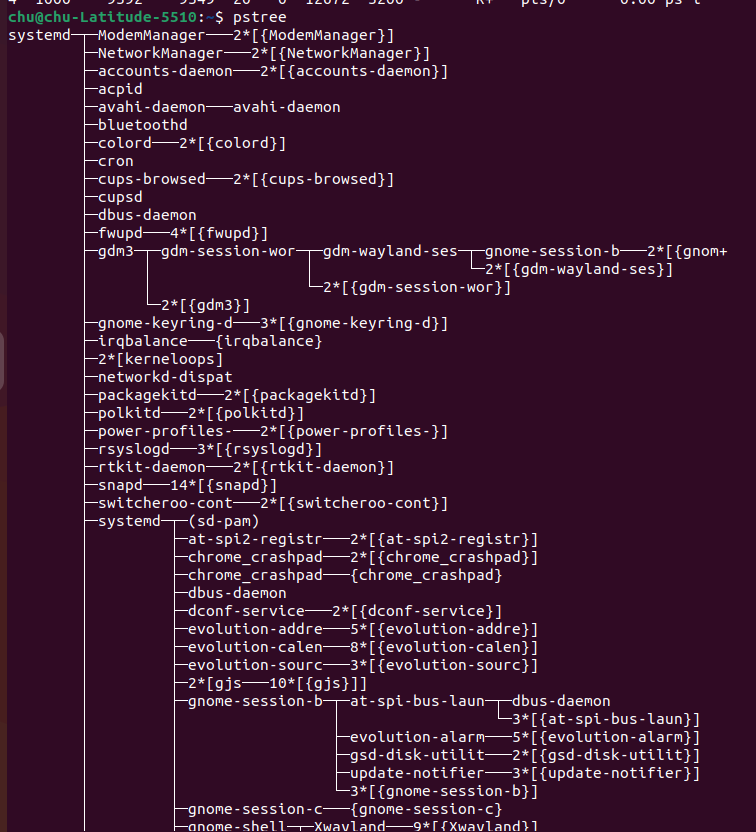






# 

# Xem danh sách các process đang chạy trên hệ thống bằng lệnh pstree



Cấu trúc cơ bản của pst

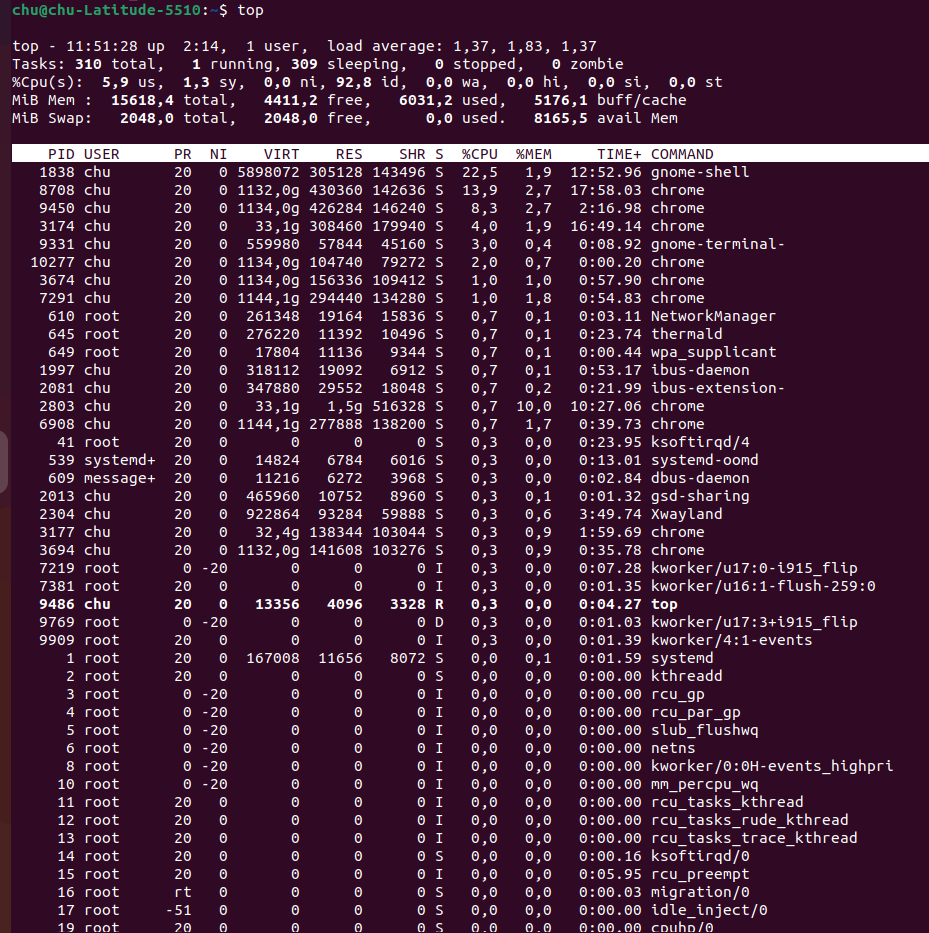
parent————child(1)————subchild (1)

| |--subchild (2)

|

|-child(2)

# Xem cách danh sách process đang chạy trên hệ thống bằng lệnh top



Dòng 1:



* Thời gian hiện tại của hệ thống
* Thời gian uptime
* Số lượng người dùng
* Trung bình tải
* Trung bình tải hiển thị thời gian load hệ thống trong 1 phút, 5 phút và 15 phút cuối.

Dòng 2:



* Tổng số tác vụ có trên máy chủ
* Số lượng tác vụ đang chạy
* Số lượng tác vụ trong trạng thái “ngủ”
* Số lượng tác vụ đã dừng
* Số lượng tác vụ zombie (tiến trình không tồn tại hoặc bị hỏng)

Dòng 3:



* %us (user cpu time): phần trăm do tiến trình của người dùng (non root) sử dụng
* %sy (system cpu time): phần trăm do tiến trình của hệ thống (root) sử dụng
* %ni (user nice cpu time): phần trăm do các tiến trình có mức độ ưu tiên thấp sử dụng
* %id (idle cpu time): phần trăm CPU đang rảnh
* %wa (io wait cpu time): phần trăm CPU để đợi trong khi các tiến trình I/O đang xử lý
* %hi (hardware irq): phần trăm để xử lý gián đoạn phần cứng
* %si (software irq): phần trăm để xử lý gián đoạn phần mềm
* %st (steal time): phần trăm do máy ảo sử dụng

Dòng 4:



* Tổng bộ nhớ hệ thống(đơn vị Kib)
* Bộ nhớ trống
* Bộ nhớ đã sử dụng
* Bộ nhớ đệm buffer cache

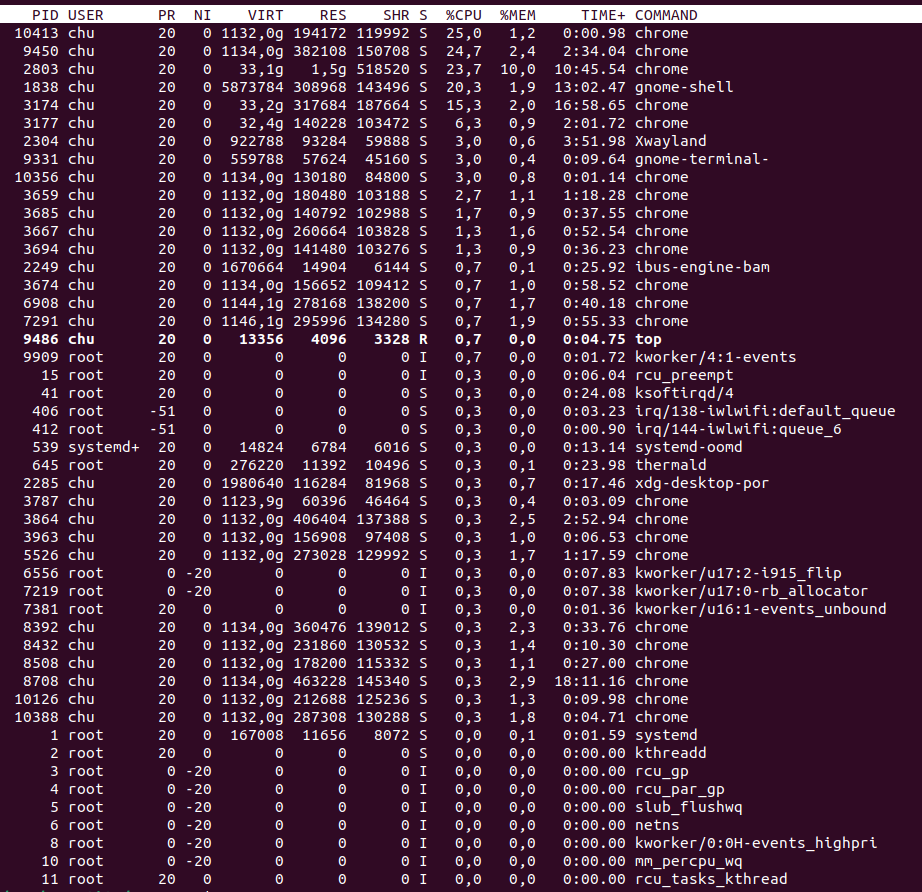
Dòng 5:



* Tổng swap có sẵn(đơn vị Kib)
* Tổng swap còn trống
* Tổng swap đã sử dụng
* Bộ nhớ khả dụng

Swap là RAM ảo, được sử dụng khi bộ nhớ vật lý (RAM) bị đầy. Luôn luôn bật Swap để khi hệ thống đầy ram vật lý sẽ không bị treo.

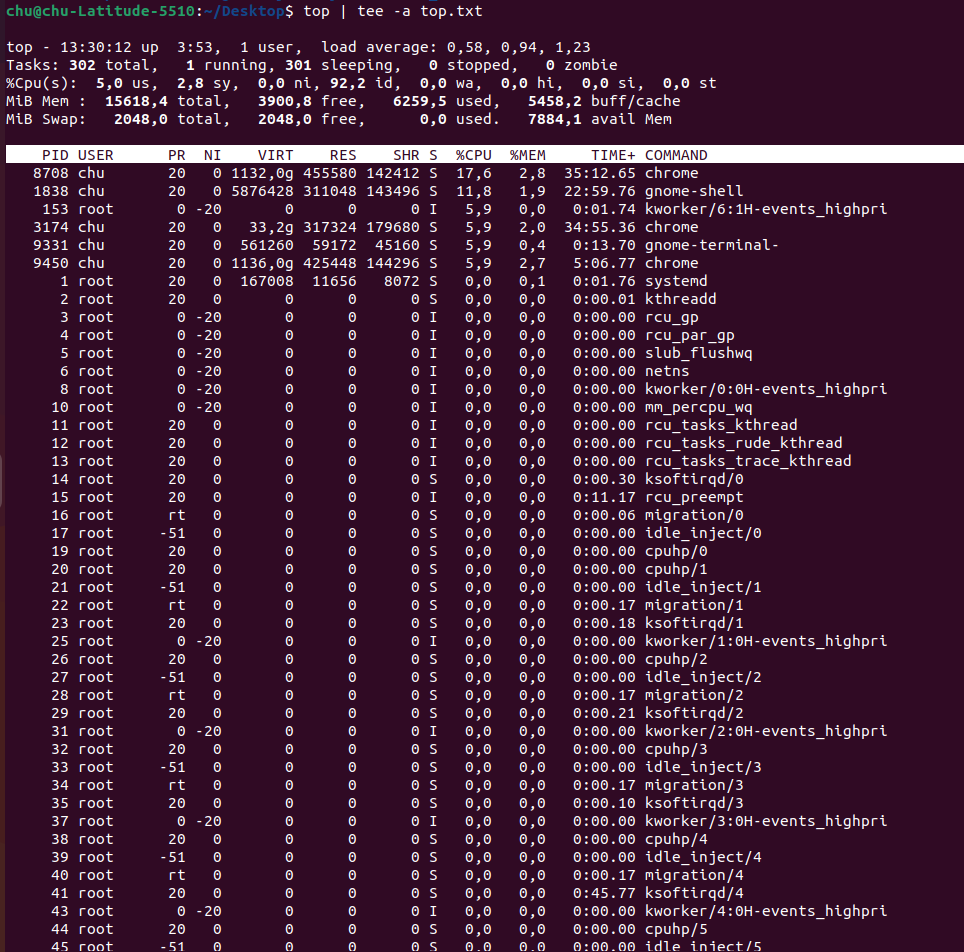
Bảng chính:



* ID tiến trình
* Người dùng(Mình có ẩn bớt một số user chứ mặc định không có trống)
* Mức độ ưu tiên
* Mức độ nice (gọi một tập lệnh shell với mức độ ưu tiên cụ thể)
* Bộ nhớ ảo được sử dụng bởi tiến trình
* Bộ nhớ “thường trú” mà một tiến trình sử dụng (tức là tiến trình luôn ở trong bộ nhớ và không thể chuyển ra thiết bị lưu trữ khác)
* Bộ nhớ có thể chia sẻ
* CPU được sử dụng bởi tiến trình theo tỷ lệ phần trăm của 1 cpu.
* Bộ nhớ được sử dụng bởi tiến trình theo tỷ lệ phần trăm
* Thời gian tiến trình đã được chạy
* Lệnh

Các tham số chính cho lệnh top

* -h – Hiển thị phiên bản hiện tại
* -c – Tham số này chuyển đổi trạng thái cột lệnh từ hiển thị lệnh sang hiển thị tên chương trình và ngược lại
* -d – Chỉ định thời gian trễ khi refresh màn hình
* -o – Sắp xếp theo trường được đặt tên
* -p – Chỉ hiển thị các tiến trình với ID được chỉ định
* -u – Chỉ hiển thị những tiến trình của người dùng được chỉ định
* -i – Không hiển thị các idle task

1. Lưu các kết quả của lệnh top vào file
2. 

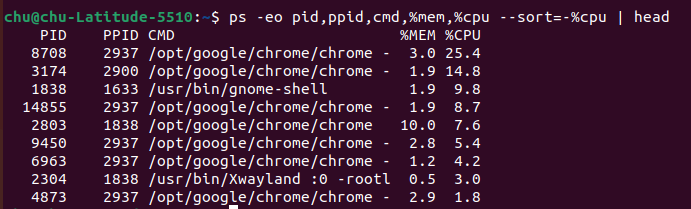
# 5. Xem tỉ lệ CPU, RAM hệ thống đang sử dụng của từng process đang chạy:

# top

O %CPU>0.0

O %MEM>0.0

# In thông tin process đang sử dụng nhiều CPU nhất



# Đếm số process đang thực thi trên máy



# Đếm số process của user root đang thực thi trên máy



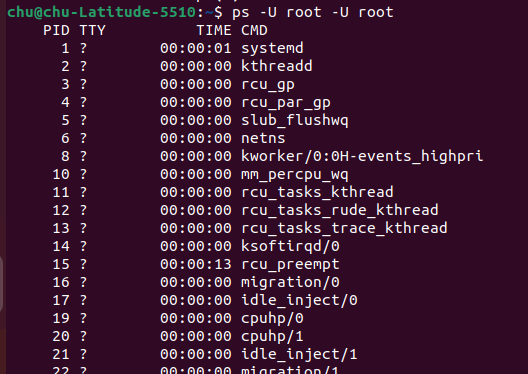
# Đếm số process “httpd” của user root đang thực thi trên máy



# Cho biết có bao nhiêu process đang chạy trong hệ thống

# tạo file





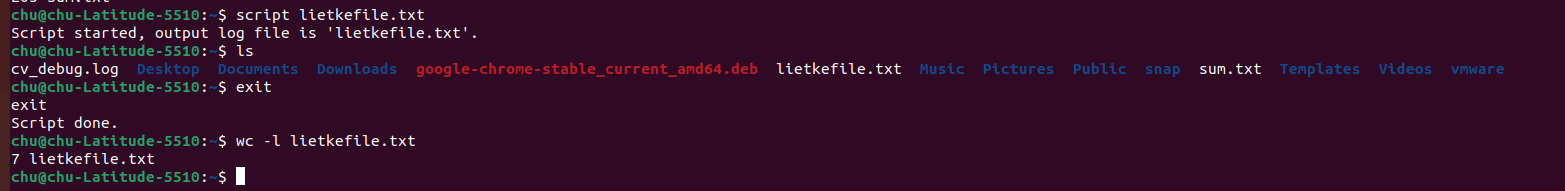
lưu file:



xem có bao nhiêu tiến trình:



# Thực thi lệnh in thông tin tất cả các tập tin, thư mục trong hệ thống. Kết quả lưu vào file



1. Chuyển lệnh trên vào chế độ bg (chế độ chạy nền)

mở lệnh top

Ctrl + z

bg

để chạy lại các file: fg