

Process

Content

Process trong linux hoạt động như nào

Các lệnh làm việc với process (bg, fg, kill, jobs)

Process trong linux hoạt động như nào?

Kernel tạo một process đầu tiên, có số hiệu tiến trình (PID) là 1, và gọi tiến trình này là "init".

Chương trình có thể khởi chạy một chương trình khác được diễn giải trong cơ chế process gọi là process cha sinh ra process con.

Kernel sẽ duy trì thông tin về mỗi một process.

Process trong linux hoạt động như nào?

Theo dõi tiến trình với **ps**

Câu lệnh: **ps [option]**

ps

```
toe@ToeUbuntu:~$ ps
  PID TTY          TIME CMD
117343 pts/0    00:00:00 bash
117521 pts/0    00:00:00 man
117529 pts/0    00:00:00 pager
117682 pts/0    00:00:00 ps
```

ps chỉ hiển thị thông tin của terminal session hiện tại

PID	Id của tiến trình
TTY	Thông tin terminal đang chạy process đó
TIME	Thời gian chiếm CPU của process tương ứng
CMD	Câu lệnh để thực hiện tiến trình đó

Process trong linux hoạt động như nào?

ps x

```
toe@ToeUbuntu:~$ ps x
  PID TTY      STAT   TIME COMMAND
    965 ?        Ss      0:00 /lib/systemd/systemd --user
    966 ?        S      0:00 (sd-pam)
    972 tty1     S+      0:00 -bash
   117341 ?
   117343 pts/0   Ss      0:00 -bash
   117344 ?
   117396 ?
   117397 ?
   117521 pts/0   T      0:00 man ls
   117529 pts/0   T      0:00 pager
   125227 ?
   125228 pts/0   R+      0:00 ps x
```

STAT mới xuất hiện, viết tắt của **state** để chỉ trạng thái của process tương ứng

Process trong linux hoạt động như nào?

Trạng thái	Ý nghĩ
R	Running. Process đang chạy hoặc sẵn sàng để chạy
S	Sleeping. Process đang đợi một event để tiếp tục chạy
D	Process đang đợi I/O
T	Stopped. Process đang trong quá trình dừng chạy
Z	Zombie process. Đây là các tiến trình con đã bị chấm dứt nhưng chưa được giải phóng bởi process cha
<	Process có độ ưu tiên cao, có thể có nhiều thời gian CPU hơn
N	Process có độ ưu tiên thấp, chỉ có thể chiếm CPU khi các process khác có độ ưu tiên cao hết thời gian CPU

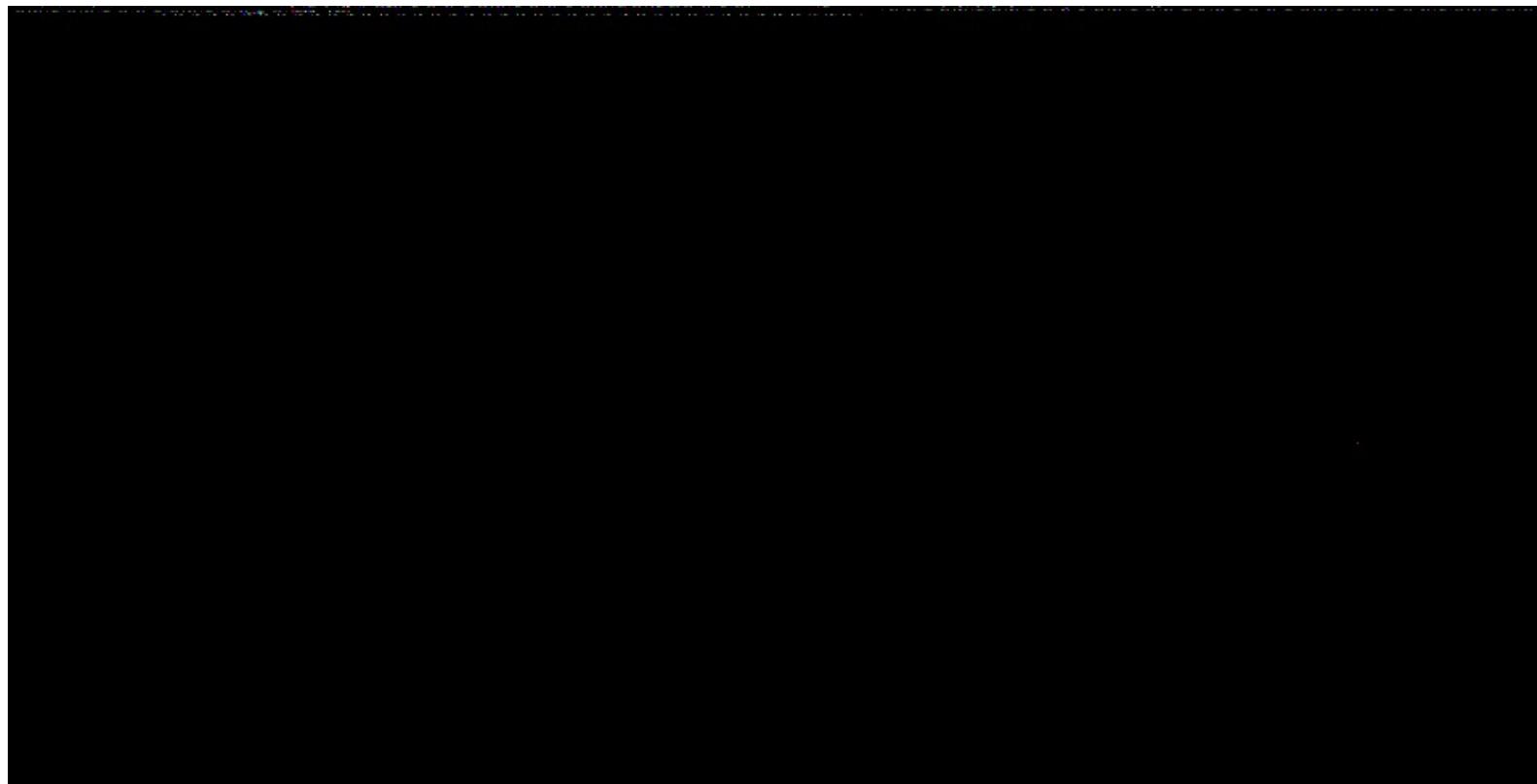
Process trong linux hoạt động như nào?

ps aux :Hiển thị process thuộc về mọi user

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.5	100776	11744	?	Ss	03:02	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S	03:02	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	03:02	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	03:02	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	03:02	0:00	[slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	03:02	0:00	[netns]
root	8	0.0	0.0	0	0	?	I<	03:02	0:00	[kworker/0:0H-events_
root	10	0.0	0.0	0	0	?	I<	03:02	0:00	[mm_percpu_wq]
root	11	0.0	0.0	0	0	?	S	03:02	0:00	[rcu_tasks_rude_]
root	12	0.0	0.0	0	0	?	S	03:02	0:00	[rcu_tasks_trace]
root	13	0.0	0.0	0	0	?	S	03:02	0:04	[ksoftirqd/0]
root	14	0.0	0.0	0	0	?	I	03:02	0:07	[rcu_sched]
root	15	0.0	0.0	0	0	?	S	03:02	0:00	[migration/0]
root	16	0.0	0.0	0	0	?	S	03:02	0:00	[idle_inject/0]
root	18	0.0	0.0	0	0	?	S	03:02	0:00	[cpuhp/0]
root	19	0.0	0.0	0	0	?	S	03:02	0:00	[kdevtmpfs]
root	20	0.0	0.0	0	0	?	I<	03:02	0:00	[inet_frag_wq]

Process trong linux hoạt động như nào?

Theo dõi tiến trình với **top**



Process trong linux hoạt động như nào?

```
top - 10:03:09 up 7:00, 2 users, load average: 0.13, 0.04, 0.01
Tasks: 116 total, 1 running, 111 sleeping, 4 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.7 sy, 0.0 ni, 98.6 id, 0.3 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1964.0 total, 1295.9 free, 266.2 used, 401.9 buff/cache
MiB Swap: 1378.0 total, 1378.0 free, 0.0 used. 1563.6 avail Mem
```

Process trong linux hoạt động như nào?

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
117344	toe	20	0	7368	3612	3352	S	0.3	0.2	0:12.28	bash
145429	toe	20	0	10476	3848	3248	R	0.3	0.2	0:02.43	top
149683	root	20	0	0	0	0	I	0.3	0.0	0:00.88	kworker/0:1-events
1	root	20	0	100776	11748	8424	S	0.0	0.6	0:01.70	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
13	root	20	0	0	0	0	S	0.0	0.0	0:04.76	ksoftirqd/0
14	root	20	0	0	0	0	I	0.0	0.0	0:08.72	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.40	migration/0
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kaudittd
22	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
24	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
25	root	20	0	0	0	0	S	0.0	0.0	0:02.82	kcompactd0
26	root	25	5	0	0	0	S	0.0	0.0	0:00.00	lru

Foreground và Background Processes

Foreground

Khi thực hiện lệnh hay tác vụ nào thì chạy trực tiếp và đợi nó hoàn thành xong mới có thể tiếp tục thực hiện một lệnh khác

Background

Là tiến trình hoặc lệnh được bắt đầu từ terminal và chạy ở chế độ nền, không có sự tương tác với người dùng.

Để chạy một lệnh trong nền, thêm dấu ‘&’ sau câu lệnh:
command &

Command

Command: fg [job_spec] - được sử dụng để đặt một công việc background sang foreground

Command: bg [job_spec] - được sử dụng để đặt một công việc foreground sang background

%number	%1, %2	Sử dụng mã số công việc
%string	%ping	Sử dụng chuỗi có tên bắt đầu bằng string
%+ hoặc %%	%+ hoặc %%	Đề cập đến công việc hiện tại
%-	%-	Đề cập đến công việc trước đó

Command: jobs - được sử dụng để liệt kê các công việc đang chạy trong foreground và background

Command

Command: kill [signal] [pid] - được sử dụng để ngắt tiến trình đang chạy bằng cơ chế tín hiệu Kill Signal

* Để hiển thị đầy đủ các Signal: kill -l

15, SIGTERM

9, SIGKILL

Thực hành

