

Λειτουργικά Συστήματα Ι

Ασκήσεις Πράξης

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών
Σχολή Μηχανικών



Διεργασίες

Διεργασίες

Όλες οι πληροφορίες κατάστασης μιας διεργασίας, οι οποίες είναι απαραίτητες για την ορθή και αποδοτική διαχείριση διεργασιών από το σύστημα, βρίσκονται σε μια δομή δεδομένων του πυρήνα που είναι γνωστή ως **PCB** (process control block).

Κάθε τέτοια δομή, άρα κάθε διεργασία χαρακτηρίζεται από ένα μοναδικό θετικό ακέραιο αριθμό που ονομάζεται αριθμός ή ταυτότητα διεργασίας (**PID** - process identifier).

ps

Η εντολή **ps** (process status) εμφανίζει πληροφορίες για τις ενεργές διεργασίες του χρήστη. Η σύνταξή της είναι:

ps [options]

Οι πιο σημαντικές επιλογές της είναι:

- e, -A** όλες τις διεργασίες του συστήματος
- f** περισσότερες πληροφορίες (full format listing)
- F** εκτεταμένη λίστα πληροφοριών (extra full format)
- u user** διεργασίες του χρήστη user
- o format** μορφή εμφάνισης οριζόμενη από τον χρήστη
- C commandlist** μόνο τις διεργασίες των οποίων το εκτελέσιμο είναι στη λίστα

ps

\$ ps

PID	TTY	TIME	CMD
541	tty1	00:00:00	bash
542	tty2	00:00:00	ps

\$ ps -ef

every, full

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	Mar13	?	...	/sbin/init
root	2	0	0	Mar13	?	...	[kthreadd]
...							
nemo	1483	1477	0	18:39	tty1	...	bash
nemo	1489	1483	0	18:40	tty1	...	ps -ef

ps

Η χρήση της επιλογής **-o** μας επιτρέπει να επιλέξουμε τις στήλες που θέλουμε να εμφανίσουμε:

```
$ ps -eo pid,user,comm,lstart
```

PID	USER	COMMAND	STARTED
1	root	systemd	Sun Dec 24 04:51:28 2023
2	root	kthreadd	Sun Dec 24 04:51:28 2023
3	root	rcu_gp	Sun Dec 24 04:51:28 2023
...			
14348	nemo	ps	Sun Dec 24 14:54:38 2023

top

- Η εντολή **top** (table of processes) εμφανίζει έναν πίνακα με τις ενεργές διεργασίες του συστήματος και άλλες χρήσιμες πληροφορίες, οι οποίες ανανεώνονται κάθε τρία δευτερόλεπτα.
- Η έξοδος της εντολής αποτελείται από δύο τμήματα. Στο επάνω μέρος εμφανίζονται στατιστικά στοιχεία για τις διεργασίες και τους πόρους του συστήματος, ενώ το κάτω μέρος αποτελείται από έναν (διαλογικό) πίνακα με τις ενεργές διεργασίες. Ο πίνακας είναι ταξινομημένος σύμφωνα με το ποσοστό χρήσης της κεντρικής μονάδας επεξεργασίας (%**CPU**).

```

top - 13:42:16 up 2:30, 1 user, load average: 0.02, 0.01, 0.00
Tasks: 70 total, 1 running, 69 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2033772 total, 1828272 free, 76368 used, 129132 buff/cache
KiB Swap: 2094076 total, 2094076 free, 0 used. 1814264 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
638	nemo	20	0	42692	3704	3148	R	0.3	0.2	0:00.17	top
1	root	20	0	56876	6712	5320	S	0.0	0.3	0:02.03	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.08	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/u256:0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.69	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.07	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
15	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
17	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kcompactd0
19	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
21	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged
22	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	crypto
23	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
24	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
25	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kblockd
26	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	devfreq_wq
27	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	watchdogd
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
29	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	vmstat
41	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kthrotld
42	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	ipv6_addrconf

- Εμφάνιση δυνατοτήτων με τα πλήκτρα **[h]** ή **[?]**.
- Για παράδειγμα, αν θέλουμε να τερματίσουμε μία διεργασία πατάμε το πλήκτρο **[k]** και στη συνέχεια πληκτρολογούμε τον αριθμό της επιθυμητής διεργασίας και πατάμε το πλήκτρο **[Enter]**.
- Για να αλλάξουμε κριτήριο ταξινόμησης ή να προβάλλουμε διαφορετικά πεδία πατάμε αρχικά το πλήκτρο **[f]** για να εμφανίσουμε τα διαθέσιμα πεδία. Χρησιμοποιώντας τώρα τα πλήκτρα με τα βέλη επιλέγουμε το επιθυμητό πεδίο και πατάμε το πλήκτρο **[s]** για να γίνει πεδίο ταξινόμησης ή το πλήκτρο **[Space]** για να επιλεγεί ή αποεπιλεγεί.
- Η μόνιμη αποθήκευση των αλλαγών με **[Shift]+[w]**.

kill

Ο τερματισμός μιας διεργασίας γίνεται με την εντολή **kill**. Πιο συγκεκριμένα, η βασική λειτουργία της εντολής είναι η αποστολή ενός σήματος σε κάποια διεργασία. Τα σήματα είναι διακοπές λογισμικού και αποτελούν έναν απλό και ασύγχρονο τρόπο επικοινωνίας μεταξύ διεργασιών.

Η σύνταξη της εντολής είναι:

kill [options] pid

Η εντολή έχει δύο μόνο επιλογές:

-signal, -s signal, --**signal** signal καθορίζει το σήμα signal που θα σταλεί

-I, -L εμφανίζει μια λίστα με τα διαθέσιμα σήματα

kill

```
$ kill -l                                # show available signals
1) SIGHUP      2) SIGINT      3) SIGQUIT  4) SIGILL
5) SIGTRAP     6) SIGABRT     7) SIGBUS    8) SIGFPE
9) SIGKILL    10) SIGUSR1    11) SIGSEGV  12) SIGUSR2
13) SIGPIPE   14) SIGALRM    15) SIGTERM
...
63) SIGRTMAX-1 64) SIGRTMAX
```

kill

```
$ ps                                # show users processes
  PID TTY          TIME CMD
  463 tty1        00:00:00 bash
 1413 tty2        00:00:00 ps
$ kill 463                         # try to kill bash with SIGTERM
$ kill -9 463                      # kill bash with SIGKILL
```

pgrep, pkill

pgrep [options] pattern

pkill [options] pattern

Οι πιο σημαντικές επιλογές τους είναι:

- s, --signal** signal καθορίζει το σήμα signal που θα σταλεί σε κάθε διεργασία που ταιριάζει με το πρότυπο pattern (pkill)
- c, --count** επιστρέφει το πλήθος των διεργασιών (pgrep)
- G, --group** group διεργασίες της ομάδας group
- U, --uid** user διεργασίες του χρήστη user
- v, --inverse** αντιστρέφει το πρότυπο ταύτισης
- x, --exact** διεργασίες που ταιριάζουν με το πρότυπο

pgrep

```
$ pgrep system      # pids name begins with system
1
178
194

$ pgrep -l systemd  # list also processes names
1 systemd

$ pgrep -U nemo -c    # count processes of a user
3

$ pgrep -v -l -U root # all users except root
336 avahi-daemon
339 dbus-daemon
```

Εκτέλεση διεργασίας στο παρασκήνιο

Η εκτέλεση μιας διεργασίας που διαρκεί μεγάλο χρονικό διάστημα έχει ως συνέπεια τη δέσμευση του τερματικού μέχρι τον τερματισμό της. Πρέπει δηλαδή να περιμένουμε να ολοκληρωθεί η διεργασία για να εμφανιστεί πάλι το σήμα προτροπής του φλοιού ώστε να πληκτρολογήσουμε την επόμενη εντολή.

Για να αποφύγουμε τη δέσμευση του τερματικού μπορούμε στείλουμε την εκτέλεση μιας εντολής στο παρασκήνιο με τον τελεστή **&** στο τέλος μιας εντολής.

Εκτέλεση διεργασίας στο παρασκήνιο

Για παράδειγμα, η εντολή **ls -R / > files** θα αργήσει να εκτελεστεί καθώς καταγράφει τα ονόματα όλων των αρχείων του συστήματος στα οποία έχει δικαίωμα πρόσβασης ο χρήστης στο αρχείο files. Για να πάρουμε άμεσα τον έλεγχο του τερματικού, την στέλνουμε για εκτέλεση στο παρασκήνιο προσθέτοντας το **&** μετά την εντολή:

```
$ ls -R / > files 2> /dev/null &      # run in bg  
[1] 7227
```

```
$                                     # ready for next command
```

Ο πρώτος αριθμός ονομάζεται αριθμός εργασίας και χαρακτηρίζει τις διεργασίες που βρίσκονται στο παρασκήνιο ή έχουν ανασταλεί προσωρινά.

jobs

Η εντολή **jobs** εμφανίζει την κατάσταση των διεργασιών που βρίσκονται στο παρασκήνιο:

```
$ jobs # display status of jobs
[1]+  Running      ls -R / > files 2> /dev/null &
```

Οι σημαντικότερες επιλογές της είναι:

- l εμφανίζει επιπλέον και τον αριθμό της διεργασίας
- p εμφανίζει μόνο τον αριθμό της διεργασίας
- r εμφανίζει μόνο τις διεργασίες που εκτελούνται
- s εμφανίζει μόνο τις διεργασίες που έχουν διακοπεί

fg, bg

Η εντολή **fg** (foreground) επαναφέρει την εκτέλεση μιας διεργασίας στο προσκήνιο. Για να επαναφέρουμε την εκτέλεση της υπ' αριθμόν 1 εργασίας στο προσκήνιο δίνουμε την εντολή:

```
$ fg %1                # move job to the foreground
```

Η εντολή **bg** (background) συνεχίζει την εκτέλεση μιας διεργασίας στο παρασκήνιο. Η επόμενη εντολή στέλνει την υπ' αριθμόν 1 εργασία στο παρασκήνιο:

```
$ bg %1                # move job to the background
```

Διακοπή διεργασίας προσκήνιου

Η διακοπή μιας διεργασίας που εκτελείται στο προσκήνιο γίνεται πατώντας τον συνδυασμό πλήκτρων **[Ctrl]+[Z]**.

```
$ ls -R /          # list all file system
bin
boot
...               # press [Ctrl]+[Z] to suspend
[1]+  Stopped      ls -R /
$
```

Τερματισμός διεργασίας παρασκηνίου

Η εντολή `kill` με όρισμα τον αριθμό εργασίας μπορεί να χρησιμοποιηθεί για τον τερματισμό μιας διεργασίας που βρίσκεται στο παρασκήνιο. Το παράδειγμα που ακολουθεί κάνει χρήση της εντολής `sleep`, η οποία χρησιμοποιείται για τη δημιουργία καθυστέρησης στα σενάρια φλοιού.

```
$ sleep 100 &                # run sleep 100 sec in bg
[1] 9415
$ kill %1                     # kill job number 1
$ jobs
[1]+  Terminated           sleep 100
```

Ερωτήσεις

