

Εισαγωγή στην Υπολογιστική Βιολογία

Διάλεξη 4η :

Unix(3) : Εντολές : grep, find, sort, wc, tar, gzip,
man (apropos), Έλεγχος εργασιών : &, fg, bg, jobs, ps, kill,
killall, top, Καθιερωμένη είσοδος / έξοδος / έξοδος διαγνωστικών
και επανακαθορισμός τους, συνδέσεις προγραμμάτων (pipes), editors.

Η εντολή grep

Σύνταξη : grep [options] <expression> <files>

Η εντολή grep ερευνά τα οριζόμενα αρχεία για την παρουσία της έκφρασης expression. Κάθε φορά που η έκφραση βρίσκεται, η grep τυπώνει την γραμμή του αντίστοιχου αρχείου που περιέχει την έκφραση.

Η εντολή grep

Παραδείγματα

```
grep 'October' my_file.txt
```

```
grep 'Escherichia' *.dat
```

```
grep '[Tt]est' /usr/local/lib/*
```

```
grep -R 'test' ./teach/
```

Η εντολή find

Σύνταξη :

```
find <dir> -name <name>
```

```
find <dir> -mtime <time>
```

```
find <dir> -name <name> -exec <command> \;
```

Με την πρώτη μορφή βρίσκουμε τη θέση (διαδρομή) των αρχείων με το όνομα name. Η έρευνα ξεκινά από τον κατάλογο dir. Με τη δεύτερη μορφή εντοπίζουμε όσα αρχεία τροποποιήθηκαν πριν από <time> ημέρες. Με την τελευταία, για κάθε ένα από τα αρχεία με το όνομα name που εντοπίζονται εκτελείται η εντολή command.

Η εντολή find

Παραδείγματα

```
find ./ -name 'old_file.txt'
```

```
find /usr/local/ -name '*[0-2]*.dat'
```

```
find ./ -mtime -2
```

```
find ./ -mtime +2
```

```
find ./ -mtime 2
```

```
find /usr/lib/ -name '*.2del' -exec rm -rf {} \;
```

```
find ./ -name '*.PREV' -exec grep -H 'test' {} \;
```

Η εντολή sort

Σύνταξη : `sort <files>`
 `sort -k <field> <files>`
 `sort -n -k <field> <files>`

Στην πρώτη μορφή τα περιεχόμενα των οριζόμενων αρχείων μπαίνουν σε αλφαβητική σειρά. Στη δεύτερη μορφή, για την ταξινόμηση χρησιμοποιείται η στήλη (το πεδίο) που ορίζεται από το field. Στην τρίτη μορφή η ταξινόμηση γίνεται αριθμητικά (και όχι αλφαβητικά).

Η εντολή wc

[w]ord [c]ount

Σύνταξη : wc <files>

Η wc τυπώνει τον αριθμό γραμμών, λέξεων και χαρακτήρων που περιέχονται σε κάθε ένα από τα οριζόμενα αρχεία.

Επίδειξη χρήσης

grep, find, sort, wc



Η εντολή tar

[t]ape [ar]chive

Σύνταξη : tar cvf <file.tar> <files>
 tar tvf <file.tar>
 tar xvf <file.tar>

Στην πρώτη μορφή δημιουργείται ένα αρχείο με το όνομα file.tar το οποίο περιέχει όλα τα οριζόμενα αρχεία ή καταλόγους (και τα περιεχόμενα τους). Στη δεύτερη μορφή τυπώνονται τα ονόματα των αρχείων και καταλόγων που περιέχονται στο file.tar. Στη τρίτη μορφή τα περιεχόμενα του file.tar ανακτώνται από το αρχείο file.tar.

Η εντολή tar

Παραδείγματα

[c]reate :

```
tar cvf test.tar mydir/
```

lis[t] :

```
tar tvf test.tar
```

e[x]tract :

```
tar xvf test.tar
```

Η εντολές gzip & gunzip

Συμπίεση και αποσυμπίεση αρχείων

Σύνταξη : gzip <files>
 gzip -r <dirs>
 gunzip <files>
 gunzip -r <dirs>

Οι εντολές gzip (για [G]nu-zip) και gunzip (για [G]nu-unzip) συμπιέζουν και αποσυμπιέζουν αρχεία.

Η εντολή man

[man]ual page

Σύνταξη : man <εντολή>
 man -k <λέξη-κλειδί>
 apropos <λέξη-κλειδί>
 man -K <λέξη-κλειδί>

Στην πρώτη μορφή τυπώνονται οι οδηγίες χρήσης της εντολής (ή θέματος) που ορίζεται (εάν υπάρχει). Στη δεύτερη μορφή τυπώνονται τα ονόματα των σελίδων βοήθειας η σύντομη περιγραφή των οποίων περιέχει τη λέξη-κλειδί. Στην τρίτη μορφή ερευνώνται όλες οι σελίδες βοήθειας για την ύπαρξη της λέξης-κλειδί.

Επίδειξη χρήσης

tar, gzip, gunzip, man



Εργασίες και έλεγχος εργασιών

Στο unix μπορείτε να σταματάτε και να επανεκκινείτε μια εργασία, να έχετε πολλές εργασίες παρασκήνιου, να βλέπετε ποιες είναι αυτές, να φέρνετε στο προσκήνιο όποια από αυτές θέλετε κοκ.

Οι βασικές μέθοδοι χειρισμού των εργασιών είναι :

CTRL-Z

Σταματάει την τρέχουσα εργασία και την βάζει στο παρασκήνιο.

jobs

Τυπώνει ένα κατάλογο με τις σταματημένες εργασίες και τις εργασίες που εκτελούνται στο παρασκήνιο.

Εργασίες και έλεγχος εργασιών

Εντολή &

Εκτελεί την 'εντολή' στο παρασκήνιο.

fg %n

Φέρνει στο προσκήνιο την εργασία με αύξων αριθμό 'n' (όπως δίνεται από την εντολή jobs).

stop %n

Σταματάει την εκτέλεση της εργασίας με αύξων αριθμό 'n' (όπως δίνεται από την εντολή jobs).

kill %n

Οριστικά διαγράφει την εργασία με αύξων αριθμό 'n' (όπως δίνεται από την εντολή jobs).

Εργασίες και έλεγχος εργασιών

Η εντολή ps

```
$ ps
  PID TTY          TIME CMD
 8821 pts/4    00:00:00 tcsh
11259 pts/4    00:00:00 xclock
11260 pts/4    00:00:00 joe
11261 pts/4    00:00:00 ps
```


Εργασίες και έλεγχος εργασιών

Η εντολή ps

```
$ ps -al
 F S      UID      PID  PPID  C PRI  NI ADDR      SZ WCHAN  TTY          TIME CMD
100 S      0         1        0  0  68   0  -       343 do_sel ?           00:00:05 init
040 S      0         2        1  0  69   0  -         0 contex ?           00:00:07 keventd
040 S      0         3        1  0  69   0  -         0 apm_na ?           00:00:00 kapid
-----
000 S      500      8770     8734  0  69   0  -       598 nanosal ?           00:00:00 wntop
000 S      500      8771     8734  0  68   0  -       603 nanosal ?           00:00:00 wminfo
000 S      500      8772     8734  0  69   0  -       605 nanosal ?           00:00:00 wmmet
000 S      500      8773     8734  0  69   0  -       727 nanosal ?           00:00:01 wncapi_temp
000 R      500      8774     8734  2  79  19  -      1684 -      ?           00:06:28 wnomload
000 S      500      8775     8734  0  69   0  -       871 nanosal ?           00:00:00 wmfire
000 S      500      8776     8734  0  69   0  -       597 nanosal ?           00:00:00 wmsn
000 S      500      8777     8734  0  69   0  -       652 do_sel ?           00:00:00 wmpager
000 S      500      8788     8758  0  75   0  -       971 rt_sig pts/0       00:00:00 tcsh
000 S      500      8803     8760  0  68   0  -       782 read_c pts/1       00:00:00 tcsh
000 S      500      8819     8763  0  69   0  -       921 read_c pts/2       00:00:00 tcsh
000 S      500      8820     8761  0  69   0  -       927 rt_sig pts/3       00:00:00 tcsh
000 S      500      8821     8762  0  68   0  -       971 read_c pts/4       00:00:00 tcsh
100 S      500      9324     8820  0  69   0  -       706 do_sel pts/3       00:00:00 ssh
000 S      500      9431     8734  0  69   0  -       651 rt_sig ?           00:00:00 tcsh
000 S      500      9450     9431  0  78  19  -       559 wait4 ?           00:00:00 firefox
000 S      500      9459     9450  0  79  19  -       565 wait4 ?           00:00:00 run-mozilla.sh
000 S      500      9464     9459  0  79  19  -      9051 do_pol ?           00:00:07 firefox-bin
040 S      500      9466     9464  0  78  19  -      9051 do_pol ?           00:00:00 firefox-bin
040 S      500      9467     9466  0  79  19  -      9051 do_pol ?           00:00:00 firefox-bin
040 S      500      9469     9466  0  79  19  -      9051 nanosal ?           00:00:00 firefox-bin
000 R      500     11263     8788  0  79   0  -        792 -      pts/0       00:00:00 ps
```

Εργασίες και έλεγχος εργασιών

Η εντολή ps

```
$ ps -elf
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	STIME	TTY	TIME	CMD
100	S	root	1	0	0	68	0	-	343	do_sel	Oct24	?	00:00:05	init
040	S	root	2	1	0	69	0	-	0	contex	Oct24	?	00:00:07	[keventd]
040	S	root	3	1	0	69	0	-	0	apm_na	Oct24	?	00:00:00	[kapmd]
.....														
000	S	glykoa	8775	8734	0	69	0	-	871	nanosl	11:52	?	00:00:00	wmfire -f 3 -i eth0
000	S	glykoa	8776	8734	0	69	0	-	597	nanosl	11:52	?	00:00:00	wmsn -t 3 -m
000	S	glykoa	8777	8734	0	69	0	-	652	do_sel	11:52	?	00:00:00	wmpager -t fire --toolti
000	S	glykos	8788	8758	0	74	0	-	971	rt_sig	11:52	pts/0	00:00:00	-tosh
000	S	glykoa	8803	8760	0	68	0	-	782	read_c	11:52	pts/1	00:00:00	-csh
000	S	glykos	8819	8763	0	69	0	-	921	read_c	11:52	pts/2	00:00:00	-csh
000	S	glykoa	8820	8761	0	69	0	-	927	rt_sig	11:52	pts/3	00:00:00	-csh
000	S	glykos	8821	8762	0	68	0	-	971	read_c	11:52	pts/4	00:00:00	-csh
100	S	glykoa	9324	8820	0	69	0	-	706	do_sel	11:55	pts/3	00:00:00	[ssh]
000	S	glykos	9431	8734	0	69	0	-	651	rt_sig	12:01	?	00:00:00	/bin/tosh -c { setenv LC
000	S	glykoa	9450	9431	0	78	19	-	559	wait4	12:01	?	00:00:00	/bin/sh /usr/local/bin/f
000	S	glykos	9459	9450	0	79	19	-	565	wait4	12:01	?	00:00:00	/bin/sh /usr/local/lib/f
000	S	glykoa	9464	9459	0	79	19	-	9051	do_pol	12:01	?	00:00:07	/usr/local/lib/firefox-0
040	S	glykos	9466	9464	0	78	19	-	9051	do_pol	12:01	?	00:00:00	/usr/local/lib/firefox-0
040	S	glykoa	9467	9466	0	79	19	-	9051	do_pol	12:01	?	00:00:00	/usr/local/lib/firefox-0
040	S	glykos	9469	9466	0	79	19	-	9051	nanosl	12:01	?	00:00:00	/usr/local/lib/firefox-0
000	R	glykoa	11284	8788	0	74	0	-	793	-	16:10	pts/0	00:00:00	ps -elf

Εργασίες και έλεγχος εργασιών

Οι εντολές kill και killall

```
$ xclock &
$ ps -el
  F S      UID      PID  PPID  C  PRI  NI ADDR      SE WCHAN  TTY          TIME CMD
100 S        0        1      0  0  68   0  -    343 do_sel ?           00:00:05 init
040 S        0        2      1  0  69   0  -     0 contex ?           00:00:07 keventd
040 S        0        3      1  0  69   0  -     0 apu_na ?           00:00:00 kapnd
040 S        0        4      1  0  79  19  -     0 ksofti ?           00:00:12 ksoftirqd_CPU0
.....
040 S      500    9467    9466  0  79  19  -    9051 do_pol ?           00:00:00 firefox-bin
040 S      500    9469    9466  0  79  19  -    9051 nanosl ?           00:00:00 firefox-bin
000 S      500   11291    8788  0  69   0  -     517 read_c pts/0       00:00:00 joe
000 S      500   11301    8821  0  69   0  -     818 do_sel pts/4       00:00:00 xclock
000 R      500   11302    8821  0  79   0  -     794 -      pts/4       00:00:00 ps

$ kill 11301
$
[1] Terminated xclock
```

Εργασίες και έλεγχος εργασιών

Οι εντολές kill και killall

```
$ xclock &
$ ps -el
 F S      UID      PID  PPID  C  PRI  NI ADDR      SE WCHAN  TTY          TIME CMD
100 S        0        1      0  0  68   0  -    343 do_sel ?           00:00:05 init
040 S        0        2      1  0  69   0  -     0 contex ?           00:00:07 keventd
040 S        0        3      1  0  69   0  -     0 apu_na ?           00:00:00 kapnd
040 S        0        4      1  0  79  19  -     0 ksofti ?           00:00:12 ksoftirqd_CPU0
.....
040 S      500    9467    9466  0  79  19  -    9051 do_pol ?           00:00:00 firefox-bin
040 S      500    9469    9466  0  79  19  -    9051 nanosl ?           00:00:00 firefox-bin
000 S      500   11291    8788  0  69   0  -     517 read_c pts/0       00:00:00 joe
000 S      500   11301    8821  0  69   0  -     818 do_sel pts/4       00:00:00 xclock
000 R      500   11302    8821  0  79   0  -     794 -      pts/4       00:00:00 ps

$ killall xclock
$
[1] Terminated xclock
```

Εργασίες και έλεγχος εργασιών

Η εντολή top

```
$ top
```

```
4:17pm up 18 days, 2:01, 5 users, load average: 0.35, 0.34, 0.29
100 processes: 93 sleeping, 7 running, 0 zombie, 0 stopped
CPU states: 0.4% user, 2.2% system, 1.5% nice, 2.3% idle
Mem: 255660K av, 243264K used, 12396K free, 0K shrd, 22304K buff
Swap: 522072K av, 14536K used, 507536K free 69568K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
8774	glykos	19	19	2468	2468	636	S N	4.8	0.9	6:40	wmonload
11313	glykos	10	0	932	932	720	R	0.9	0.3	0:00	top
1	root	8	0	112	68	56	S	0.0	0.0	0:05	init
2	root	9	0	0	0	0	SW	0.0	0.0	0:07	keventd
3	root	9	0	0	0	0	SW	0.0	0.0	0:00	kapnd
4	root	19	19	0	0	0	SWN	0.0	0.0	0:12	ksoftirqd_CPU0
5	root	9	0	0	0	0	SW	0.0	0.0	0:08	kswapd
6	root	9	0	0	0	0	SW	0.0	0.0	0:00	bdflush
7	root	9	0	0	0	0	SW	0.0	0.0	0:00	kupdated
8	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	mdrecoveryd
9	root	9	0	0	0	0	SW	0.0	0.0	0:05	kjournald
10	root	9	0	0	0	0	SW	0.0	0.0	0:00	omfs_main_serve
11	root	9	0	0	0	0	SW	0.0	0.0	0:00	omfs_gc
12	root	9	0	0	0	0	SW	0.0	0.0	0:00	om_migd
13	root	9	0	0	0	0	SW	0.0	0.0	2:36	om_infoD
14	root	9	0	0	0	0	SW	0.0	0.0	0:42	memsorter
88	root	9	0	0	0	0	SW	0.0	0.0	0:00	khubd
982	root	9	0	208	160	160	S	0.0	0.0	0:02	syslogd
987	root	9	0	196	180	180	S	0.0	0.0	0:00	klogd
1007	rpc	9	0	132	52	52	S	0.0	0.0	0:00	portmap
1035	rpcuser	9	0	96	0	0	SW	0.0	0.0	0:00	rpc.statd
1097	root	9	0	752	648	640	S	0.0	0.2	0:00	apcuped

Επίδειξη χρήσης

Εργασίες και έλεγχος εργασιών



Καθιερωμένη είσοδος/έξοδος

- Ένα πρόγραμμα σε C μπορεί να διαβάσει δεδομένα είτε από κάποιο αρχείο ή από την 'καθιερωμένη είσοδο' (standard input) η οποία (απουσία επανακαθορισμού) αντιστοιχεί στο πληκτρολόγιο.
- Αντίστοιχα, ένα πρόγραμμα μπορεί να εξάγει τα αποτελέσματα του είτε σε ένα αρχείο ή στην 'καθιερωμένη έξοδο' (standard output) η οποία (απουσία επανακαθορισμού) αντιστοιχεί στο τερματικό (οθόνη).
- Για την έξοδο των διαγνωστικών μηνυμάτων (π.χ. σφαλμάτων) υπάρχει η 'καθιερωμένη έξοδος σφαλμάτων' (standard error) η οποία (απουσία επανακαθορισμού) επίσης αντιστοιχεί στο τερματικό.

Επανακαθορισμός εισόδου/εξόδου

Το unix παρέχει τη δυνατότητα επανακαθορισμού των καθιερωμένων εισόδων/εξόδων με τη χρήση των συμβόλων '>' και '<' :

- Το σύμβολο '>' επανακαθορίζει την καθιερωμένη έξοδο.
- Το σύμβολο '<' επανακαθορίζει την καθιερωμένη είσοδο.

Επανακαθορισμός εισόδου/εξόδου

Παραδείγματα

```
find ./ -name '*.dat' > files.list
```

```
ls -R /usr/local > ~/list.dat
```

```
wc < test.txt
```

Συνδέσεις προγραμμάτων

Εάν θέλουμε να εκτελέσουμε διαδοχικά δύο προγράμματα έτσι ώστε η έξοδος από το πρώτο να χρησιμοποιηθεί ως είσοδος για το δεύτερο, μπορούμε να χρησιμοποιήσουμε το :

```
prog1 > temp
```

```
prog2 < temp
```

Το unix παρέχει έναν αυτόματο τρόπο για να γίνει αυτό, τα pipes :

```
prog1 | prog2
```

όπου η καθιερωμένη έξοδος του prog1 γίνεται (αυτόματα) η καθιερωμένη είσοδος του prog2.

Συνδέσεις προγραμμάτων

Παραδείγματα

```
ls | wc
```

```
grep 'test' * | wc
```

```
tar tvf myfiles.tar | grep 'thisfile.dat'
```

Editors

- vi, vim
- nedit
- xedit
- joe
- pico
- jed
- emacs
- xemacs

Επίδειξη χρήσης

Είσοδος/έξοδος/συνδέσεις, editors

