



Λειτουργικά Συστήματα

Ενότητα 5: Διαχείριση Διεργασιών στο ΛΣ Linux.
Κανονικές Εκφράσεις. Καταγραφή των εντολών
με το script. Έλεγχος παραμέτρων γραμμής
εντολών.

Επιβλέπων: Δασυγένης Μηνάς
Παυλίδου Ελένη

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org

Εργαστήριο Λειτουργικών Συστημάτων

<http://arch.ece.uowm.gr/courses/os/>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Η εντολή ps (1/5)

- Η `ps` είναι μια εντολή που επιστρέφει μια λίστα από διεργασίες καθώς και το ID τους, TTY (terminal type), `time`, `command`. Αν χρησιμοποιηθεί σκέτη δίνει μια λίστα με τις διεργασίες που τρέχουν από τον χρήστη. Αν θέλουμε να δούμε όλες τις διεργασίες που τρέχουν στο σύστημα, τότε χρησιμοποιούμε την: `ps -A` ή `ps -e`
- Η `ps` χρησιμοποιείται συχνά για να φιλτράρουμε μία ή μια ομάδα διεργασιών.



Η εντολή ps (2/5)

- Για παράδειγμα: Η `ps -p pid` φτιάχνει λίστα με τη διεργασία με ID `pid`.
Αν θέλουμε παραπάνω από μία διεργασία αυτό μπορεί να γίνει σε μια γραμμή έτσι :

```
ps -p ID1, ID2, ID3, .....
```

Ή εναλλακτικά μπορούμε να ψάξουμε μια εργασία με βάση το όνομα της και όχι το ID: `ps -C <command>`



Η εντολή ps (3/5)

- Παρακάτω είναι μια λίστα με άλλες χρήσιμες εφαρμογές της εντολής ps :
- `ps -G <groupname>` εμφανίζει τις διεργασίες που τρέχουν από το group `groupname` .
- `ps -G "accounts"` εμφανίζει τις διεργασίες που τρέχουν από το account group `accounts` .
- `ps -t <sessionid>` : φιλτράρει με βάση το terminal type .



Η εντολή ps (4/5)

- `ps -s <sessionid>` : φιλτράρει με βάση το `sessionid`.
- `ps -U "user"` : δείχνει τις διεργασίες που εκτελούνται από τον χρήστη `user`.
- `ps -Af` : Εμφανίζει όλες τις διεργασίες στο σύστημα με όλες τις λεπτομέρειες.
- `ps -a` : ψάχνει το σύνολο των οδηγιών χρήσης του συστήματος (`manual pages`) για αναφορές πάνω στην λέξη "κλειδί" που δίνεται.



Η εντολή ps (5/5)

- `ps -x`: Εμφανίζονται όλες οι διεργασίες ανεξάρτητα από το αν έχουν ξεκινήσει από τερματικό ή όχι.
- `ps -w` : Απομακρύνει το όριο εμφάνισης στο όνομα της διεργασίας.

(Η εντολή **ps** είναι παρόμοια με τη εντολή `tasklist` των Microsoft Windows.

Στο Windows PowerShell, η `ps` είναι ένα προκαθορισμένο ψευδώνυμο για το cmdlet `Get-Process`, το οποίο έχει την ίδια λειτουργία.)



Οι εντολές kill και killall (1/2)

- Όταν θέλουμε να «σκοτώσουμε» δηλαδή να τερματίσουμε μια διεργασία χρησιμοποιούμε την εντολή `kill`.

Σύνταξη: `kill -9 pid`

Όπου `pid` σημαίνει process identification number και είναι ουσιαστικά ο αριθμός ταυτότητα της διεργασίας που εμφανίζεται στην έξοδο της `ps`.

- Παράδειγμα: `kill -9 6465`



Οι εντολές kill και killall (2/2)

➤ `killall <όνομα εφαρμογής>`

Η εντολή **killall** «σκοτώνει» την εφαρμογή με το όνομά της.

❖ Προσοχή! Για να δουλέψει η εντολή θα πρέπει να δώσουμε το ακριβές όνομα του εκτελέσιμου της εφαρμογής.

▪ Για παράδειγμα, το όνομα του εκτελέσιμου της εφαρμογής `firefox` μπορεί να είναι `firefoxbin`. Η εντολή θα ήταν:

```
killall firefoxbin
```



Η εντολή `top`

Όπως είδαμε πιο πάνω, μπορούμε να χρησιμοποιήσουμε την εντολή `kill -9` μαζί με τον αριθμό ταυτότητας μιας διεργασίας (`pid`) για να τερματίσουμε μια διεργασία.

Προκειμένου να βρούμε αυτόν τον αριθμό (`pid`) χρησιμοποιούμε την εντολή `top`.

- Η εντολή `top` εμφανίζει σε πραγματικό χρόνο πληροφορίες για τις διεργασίες που εκτελούνται.



Εντολή jobs (1/2)

- Αν βρισκόμαστε σε μια σύγχυση δραστηριοτήτων έχοντας ξεκινήσει πολλές διεργασίες, ο απλούστερος τρόπος να επανεξετάσουμε τι έχουμε κάνει ως τώρα είναι να πληκτρολογήσουμε: **jobs**
- Η εντολή **jobs** προβάλλει μια λίστα των ανεσταλμένων εργασιών και των εργασιών που εκτελούνται στο παρασκήνιο στο τερματικό στο οποίο δουλεύουμε. Αν δουλεύουμε σε δυο τερματικά ταυτοχρόνως, το καθένα θα έχει τη δική του λίστα εργασιών.



Εντολή jobs (2/2)

Το αποτέλεσμα της εντολής **jobs** έχει την εξής μορφή:

```
[1] + Stopped vi report
[2] - Stopped vi test
[3] Stopped vi find . -name core -print
[4] Running ls -R /|more
```

Η πρώτη στήλη είναι ο εκχωρούμενος αριθμός εργασίας μέσα σε αγκύλες. Η δεύτερη περιέχει ένα + για την τρέχουσα εργασία και ένα – για την επόμενη «περισσότερο» τρέχουσα εργασία. Η τρίτη στήλη λέει αν η εργασία έχει ανασταλεί ή εκτελεστεί στο παρασκήνιο και η τελευταία δίνει το όνομα της εργασίας.



Εντολές Ctrl-z, fg και bg (1/2)

- Μπορούμε να σταματήσουμε την τρέχουσα εργασία , πληκτρολογώντας: **Ctrl-z**
- Όταν θέλουμε να ξαναρχίσουμε την εργασία, πληκτρολογούμε: **fg** από το **fore-ground**
ή στο παρασκήνιο **bg** από το **back-ground**
και η εργασία μας συνεχίζεται από το σημείο όπου σταμάτησε.



Εντολές Ctrl-z, fg και bg (2/2)

- Αν επεξεργαζόμασταν ένα αρχείο στον vi, θα επιστρέψαμε στη γραμμή επί της οποίας εργαζόμασταν.
- Αν σταματήσουμε μια εργασία και την ξεχάσουμε, τότε όταν επιχειρήσουμε να αποσυνδεθούμε, το σύστημα θα μας πει:
~There are stopped jobs~
και δε θα μας επιτρέπει να κάνουμε logout. Όμως αν επιχειρήσουμε επιμόνως να αποσυνδεθούμε δίνοντας πολλές φορές την εντολή logout, το σύστημα θα μας το επιτρέψει τελικά και θα «σκοτώσει» τις ανεσταλμένες εργασίες.



Αποσφαλματωτής gdb

- Ο αποσφαλματωτής (Debugger), **GNU** που συνήθως αποκαλείται **GDB**, με το εκτελέσιμό του να έχει το όνομα *gdb*, είναι ο αποσφαλματωτής του συστήματος λογισμικού GNU. Εκτελείται σε πολλά συστήματα Unix και λειτουργεί με διάφορες γλώσσες προγραμματισμού όπως η Ada, η C, η C++, κ.α.
- Το πρόγραμμα εκτελείται δίνοντας την εντολή `gdb` στο τερματικό. Για να διακόψουμε πατάμε `q` ή `quit`.
- ❖ Δώστε την εντολή `help` για να δείτε τις λίστες με τις εντολές που μπορεί να σας φανούν χρήσιμες.



Κανονικές εκφράσεις ή RE (1/3)

- Οι κανονικές εκφράσεις μπορούν να χρησιμοποιηθούν σε εντολές και προγράμματα του UNIX για:
 - Να προσδιορίσουν ονόματα αρχείων.
 - Να ανιχνεύσουν περιεχόμενα αρχείων.
 - Να μεταβάλλουν περιεχόμενα αρχείων.
- Κάθε εντολή του UNIX διαφέρει όσον αφορά τον τύπο των RE που υποστηρίζει. Ο προσδιορισμός των RE που υποστηρίζονται από κάθε εντολή είναι προτιμότερο να γίνεται ελέγχοντας την εντολή μέσω της εντολής `man` (manual pages).



Κανονικές εκφράσεις ή RE (2/3)

- Οι κανονικές εκφράσεις χρησιμοποιούνται κυρίως για την αναζήτηση αλφαριθμητικών γενικής μορφής σε ένα αρχείο. Π.χ. Για την εύρεση όλων των γραμμών ενός αρχείου που περιέχουν ένα από τα παρακάτω:
 - Τη λέξη Unix.
 - Τη λέξη UNIX.
 - Ένα πρότυπο (pattern) που αποτελείται από 4 ψηφία.
 - Έναν ταχυδρομικό κώδικα.
 - Ένα όνομα.
 - Όλα τα φωνήεντα σε αλφαβητική σειρά.



Κανονικές εκφράσεις ή RE (3/3)

- Μια κανονική έκφραση είναι μια σειρά από κανονικούς χαρακτήρες και ειδικούς τελεστές. Οι κανονικοί χαρακτήρες περιλαμβάνουν το σύνολο των πεζών και κεφαλαίων γραμμάτων, τα ψηφία, και άλλους συχνά χρησιμοποιούμενους χαρακτήρες : `~ , ' , # , _ , - , = , : , ; , , , /`
Οι ειδικοί τελεστές είναι `\ , . , * , [, ^ , $,]`.

- ❖ Περισσότερα αναφέρονται και στο εργαστηριακό φυλλάδιο!



Η εντολή grep (1/4)

- Μερικές από τις πλέον ισχυρές εντολές (λειτουργίες) του UNIX, όπως `grep` και `sed`, χρησιμοποιούν κανονικές εκφράσεις.
- Οι εντολές `grep`, `egrep`, `fgrep` χρησιμοποιούνται για αντιστοίχιση ενός προτύπου χαρακτήρων.
Για την ομαλή λειτουργία οι κανονικές εκφράσεις πρέπει να περικλείονται σε οριοθέτες (π.χ. `` ``) ώστε ο φλοιός να διερμηνεύσει τους χαρακτήρες που περιέχονται στην κανονική έκφραση ως “grep special characters” και όχι ως “shell special characters”.



Η εντολή grep (2/4)

➤ Μερικές χρήσιμες παράμετροι του `grep` είναι:

- `-c` Εμφανίζει το πλήθος των γραμμών που ταιριάζουν με το πρότυπο.
- `-i` Αγνοεί τη διάκριση μεταξύ πεζών και κεφαλαίων γραμμάτων.
- `-l` Εμφανίζει τα ονόματα των αρχείων που ταιριάζουν με το πρότυπο.
- `-n` Περιλαμβάνει τους αριθμούς γραμμών.
- `-v` Εμφανίζει τις γραμμές που δεν αντιστοιχούν στο πρότυπο.
- `-w` Βρίσκει μόνο ολόκληρες λέξεις.



Η εντολή `grep` (3/4)

- ❖ Παραδείγματα χρήσης του `grep` σε απλό κείμενο και σε κείμενο που περιέχει ειδικούς χαρακτήρες και άλλα.
- `grep unix file` (αντιστοιχεί όλες τις εμφανίσεις της λέξης `unix` από το αρχείο `file`)
- `grep '[uU]nix' file` (αντιστοιχεί `unix` και `Unix`)
- Όταν θέλουμε να εμφανίσουμε αποκλειστικά τις γραμμές που αρχίζουν με `"root"`: `grep ^root /etc/passwd`
- Αν θέλουμε να δούμε ποιοι λογαριασμοί δε διαθέτουν κάποιο φλοιό, αναζητούμε τις γραμμές που τελειώνουν με `:"` :
`grep :$ /etc/passwd`



Η εντολή grep (4/4)

➤ Χρήσιμοι συμβολισμοί.

- `c` Αντιστοιχεί στον χαρακτήρα `c`.
- `\c` Αναγκάζει το `c` να διαβαστεί μόνο ως το γράμμα.
- `^` Ταυτοποιείται κατά την αναζήτηση με την αρχή της γραμμής.
- `$` Ταυτοποιείται κατά την αναζήτηση με το τέλος της γραμμής.
- `.` Ταυτοποιείται με ένα και μόνο χαρακτήρα.
- `[xy]` Κάθε μεμονωμένος χαρακτήρας που περιέχεται στο σύνολο.
- `[^xy]` Κάθε μεμονωμένος χαρακτήρας που δεν περιέχεται στο σύνολο (exclusion set).



Παραδείγματα χρήσης του grep (1/2)

- Όλες οι γραμμές που δεν ξεκινούν από κεφαλαίο αγγλικό χαρακτήρα :
`grep '^^[A-Z]' file`
- Όλες οι γραμμές που περιέχουν `!`, `&`, `*` :
`grep '([\!*\&])' file`
- Όλες οι γραμμές με μήκος 2 χαρακτήρες :
`grep '^..$' file` ή `grep '^.\{2\}$'`
- Όλες οι γραμμές που έχουν μήκος ακριβώς 17 χαρακτήρες:
`grep '^.\{17\}$' file`
- Όλες οι γραμμές που έχουν μήκος τουλάχιστον 25 χαρακτήρες:
`grep '^.\{25,\}$'`
- Όλες οι γραμμές που δεν έχουν μήκος 3 χαρακτήρες:
`grep -v '^...$' file`



Παραδείγματα χρήσης του grep (2/2)

- Οποιοσδήποτε προσημασμένος ακέραιος: `[+\-] [0-9] [09]*`
- Όλες οι γραμμές που ξεκινούν με * : `^*`
- Όλες οι γραμμές που δεν περιέχουν αριθμούς : `^[^0-9]*$`
- Όλες οι γραμμές που περιέχουν τα έτη 1991 έως 1995 :
`199[1-5]`
- Οποιαδήποτε ακολουθία χαρακτήρων δεν περιέχει ψηφία:
`[A-Za-z][A-Za-z]*`
- Εμφάνιση όλων των directories του current directory στα οποία οι υπόλοιποι χρήστες (other) έχουν δικαίωμα execute:
`ls -l | grep '^d.....x'`



Η εντολή `sleep` (1/3)

- Η εντολή `sleep` εισάγει μια αναμονή, για όσα δευτερόλεπτα της δώσουμε. Πληκτρολογήστε:

```
% sleep 10
```

Η παραπάνω εντολή περιμένει 10 δευτερόλεπτα πριν επιστρέψει η προτροπή του κελύφους. Μέχρι να δείτε ξανά την προτροπή δεν μπορείτε να κάνετε τίποτα εκτός από το να περιμένετε.



Η εντολή `sleep` (2/3)

➤ Για να εκτελέσουμε την `sleep` στο παρασκήνιο, γράφουμε:

```
% sleep 10 &  
[1] 6259
```

Με το `&` η εντολή εκτελείται στο παρασκήνιο και η προτροπή επιστρέφει αμέσως. Μπορούμε να εκτελέσουμε άλλα προγράμματα καθώς περιμένουμε αυτή την εντολή να τελειώσει.



Η εντολή sleep (3/3)

- Η πρώτη γραμμή στο παραπάνω παράδειγμα, έχει πληκτρολογηθεί από τον χρήστη. Η επόμενη γραμμή που αναφέρει τον αριθμό της διεργασίας και της εργασίας, προέρχεται από το μηχάνημα. Ο χρήστης ειδοποιείται για τον αριθμό της εργασίας (ξεκινάει από τον αριθμό 1) ο οποίος περικλείεται σε αγκύλες και για τον αριθμό της διεργασίας. Επίσης ειδοποιείται όταν τελειώσει η εκτέλεση μιας διεργασίας που εκτελείται στο παρασκήνιο. Η εκτέλεση στο παρασκήνιο είναι χρήσιμη για διεργασίες που κάνουν αρκετό χρόνο να ολοκληρωθούν.



Το πρόγραμμα script

- Η εντολή `script` καταγράφει στο αρχείο <όνομα_αρχείου>, όλες τις εντολές που δίνονται στο τερματικό. Η σύνταξη της εντολής είναι: `script <όνομα_αρχείου>`
Αν δε δοθεί όνομα αρχείου η καταγραφή γίνεται στο αρχείο `typescript`. Η καταγραφή σταματάει όταν γραφεί `exit` στην γραμμή εντολών του UNIX.
- Η παράμετρος `-t` (`script -t`) ενεργοποιεί την καταγραφή του χρόνου.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Πανεπιστήμιο Δυτικής Μακεδονίας