

Λειτουργικά Συστήματα: Εργαστηριακή Άσκηση 1

Εβδομάδα 16-20 Οκτωβρίου 2017

Εισαγωγή στο UNIX

Το UNIX αποτελείται από καταλόγους (directories) και αρχεία (files) στα οποία αποθηκεύεται μια ποικιλία πληροφοριών, όπως είναι οι πληροφορίες για τη διαδικασία εγκατάστασης, ρυθμίσεις για τη διευθέτηση του συστήματος, προγράμματα και αρχεία με δεδομένα. Οι κατάλογοι συνιστούν ένα δένδρο με υποκαταλόγους που κρέμονται κάτω από καταλόγους που βρίσκονται σε υψηλότερο επίπεδο. Στο σχήμα 1.1 δείχνονται σχηματικά όλοι οι κατάλογοι που είναι ένθετοι μέσα στο βασικό κατάλογο (/) και που είναι διαθέσιμοι κατά την εκκίνηση του συστήματος.

Το UNIX διαθέτει μια σειρά προγραμμάτων συστήματος που μπορούν να καλούνται με εντολές του φλοιού (shell) του συστήματος, δηλαδή του προγράμματος μετάφρασης εντολών (command interpreter) που διαθέτει κάθε λειτουργικό σύστημα. Στη συνέχεια θα παρουσιασθούν οι κυριότερες από τις εντολές αυτές που έχουν σχέση με τη διαχείριση των καταλόγων και αρχείων, την επεξεργασία κειμένου, τον έλεγχο της κυριότητας και αδειών χρήσης του συστήματος και την εκτέλεση προγραμμάτων. Κάθε χρήστης μετά τη σύνδεση του με το σύστημα μεταφέρεται στον αρχικό του κατάλογο (home directory), όπου φυλάσσονται τα προσωπικά του αρχεία και οι ρυθμίσεις του.

Εντολές χρήσης καταλόγων και αρχείων

- Εντολή ls: λήψη των ονομάτων της λίστας περιεχομένων ενός καταλόγου. Με την προσθήκη της σημαίας -l μπορεί κανείς να πάρει μαζί με τη λίστα των ονομάτων των αρχείων και πληροφορίες σχετικές με άδειες πρόσβασης στο αρχείο, την ημερομηνία της τελευταίας τροποποίησης του, το μέγεθος, την κυριότητα και τη συμμετοχή στην ομάδα και την ώρα της τελευταίας τροποποίησης. Με την προσθήκη της σημαίας -a λαμβάνεται ολόκληρη η λίστα των αρχείων ενός καταλόγου, μαζί με τα κρυφά αρχεία.
- Εντολή cd: μεταφέρει τον έλεγχο από τον τρέχοντα κατάλογο στον κατάλογο που καθορίζει ο χρήστης. Για να μεταφερθεί ο χρήστης σε συγκεκριμένο κατάλογο πρέπει να πληκτρολογήσει στο προτρεπτικό του φλοιού την εντολή cd και το όνομα του καταλόγου εφόσον θέλει να μετακινηθεί κάτω από τον τρέχοντα κατάλογο στο δένδρο των καταλόγων, π.χ

```
Bash-2.3% cd Projects
```

Για να μετακινηθεί κάποιος ένα επίπεδο πάνω από τον τρέχοντα κατάλογο αρκεί να πληκτρολογήσει την εντολή με την εξής μορφή:

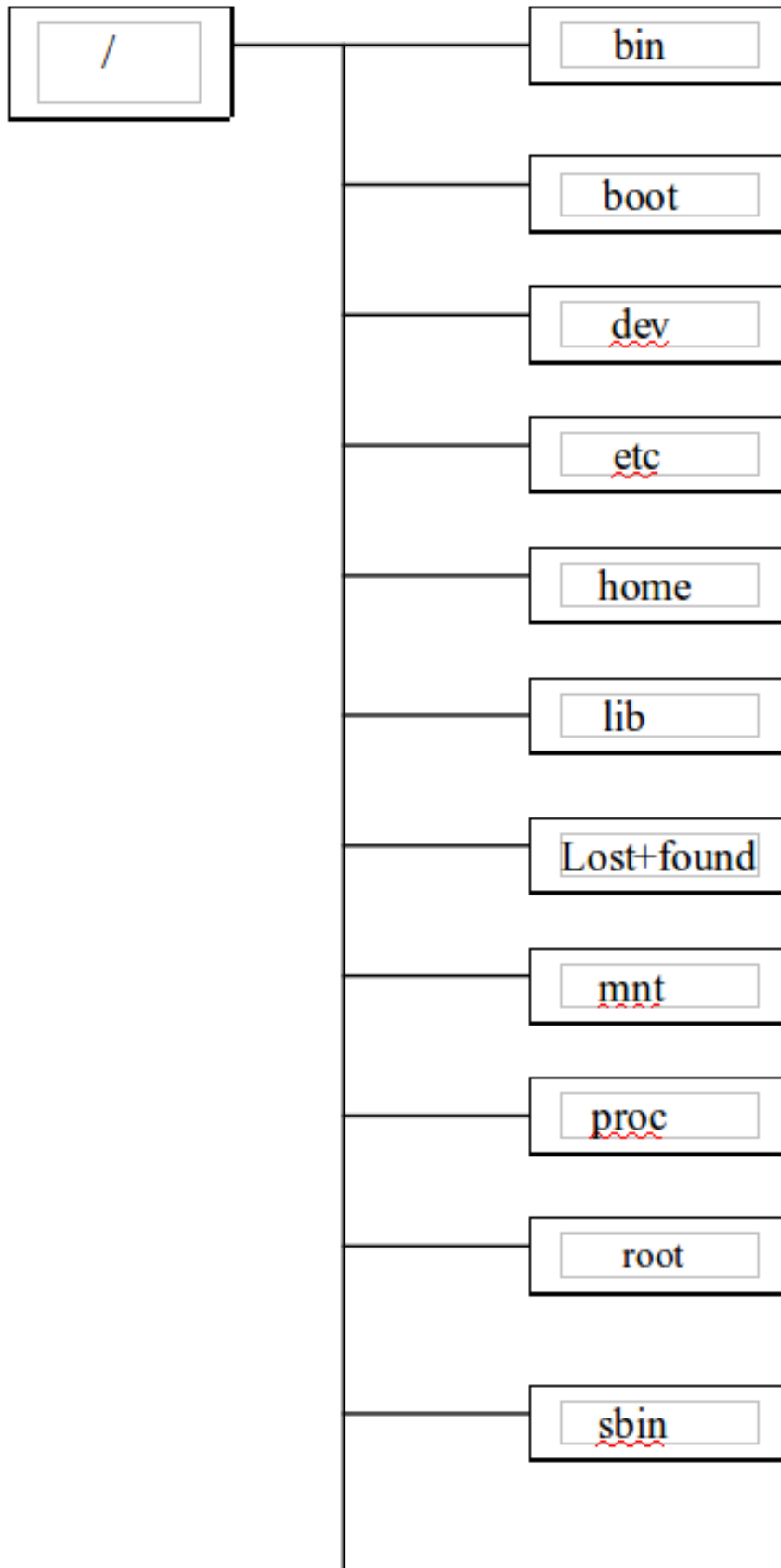
```
cd ..
```

Για να ψάξετε να βρείτε κατάλογο που βρίσκεται κάτω από τον αρχικό, π.χ. τον κατάλογο etc, αρκεί να πληκτρολογήσετε την εντολή:

```
cd /etc
```

Για να επιστρέψει κανείς στον αρχικό του κατάλογο από οποιοδήποτε σημείο που μπορεί να βρίσκεται στο δένδρο των καταλόγων αρκεί να πληκτρολογήσει cd.

Αν θέλετε να μετακινηθείτε π.χ στον κατάλογο Urgent που βρίσκεται κάτω από τον κατάλογο Projects στον αρχικό σας κατάλογο τότε αυτό μπορεί να γίνει γράφοντας όλη τη διαδρομή από τον αρχικό σας κατάλογο, δηλαδή



```
cd / home / users / y / yourid / Projects /Urgent
```

Στα περισσότερα συστήματα UNIX αντί να γραφεί ολόκληρη η διαδρομή από τον αρχικό κατάλογο μέχρι τον κατάλογο κατάληξης μπορεί εναλλακτικά να χρησιμοποιηθεί το σύμβολο '~' για να υποκαταστήσει τη διαδρομή μέχρι τον κατάλογο Projects, δηλαδή μπορεί να δοθεί η εντολή

```
cd ~/ Projects /Urgent
```

Έχετε υπόψη σας ότι ο αρχικός σας κατάλογος (home) είναι διαφορετικός από το βασικό κατάλογο (root). Θεωρήστε τον αρχικό σας κατάλογο ως ένα μικρό τομέα του συστήματος UNIX που μπορείτε να εξουσιάζετε.

- Εντολή pwd: εμφανίζει τη διαδρομή και το όνομα του τρέχοντος καταλόγου. Η διαδρομή περιέχει τα ονόματα όλων των καταλόγων που βρίσκονται πάνω από τον τρέχοντα κατάλογο δίνοντας την πλήρη εικόνα του που βρίσκεστε σε σχέση με το βασικό κατάλογο του συστήματος.

- Εντολή more: εμφανίζει τα περιεχόμενα των αρχείου σελίδα προς σελίδα. Π.χ με

```
more fortunes
```

εμφανίζεται η πρώτη σελίδα του αρχείου fortunes. Πατώντας 'ENTER' εμφανίζεται η επόμενη γραμμή του αρχείου, πληκτρολογώντας 'διάστημα (space bar)' εμφανίζεται το περιεχόμενο της επόμενης σελίδας ενώ πληκτρολογώντας b επαναφέρουμε στην οθόνη την προηγούμενη σελίδα. Επιστροφή στην προτροπή του κελύφους γίνεται με το γράμμα q.

- Εντολή cat: εμφανίζει και η εντολή αυτή το περιεχόμενο ενός αρχείου, όχι όμως σελίδα προς σελίδα . Αν το μήκος των περιεχομένων είναι μεγαλύτερο από το μήκος μιας οθόνης τότε τα περιεχόμενα θα τρέξουν πολύ γρήγορα και το μόνο που θα εμφανισθεί στην οθόνη θα είναι οι τελευταίες γραμμές του αρχείου. Όμως με την εντολή cat μπορεί να δει κανείς τα περιεχόμενα πολλών αρχείων. Π.χ με

```
cat newer.program newest.program
```

πρώτα θα περάσουν γρήγορα από την οθόνη τα περιεχόμενα του newer.program και μετά του newest.program. Η εντολή cat μπορεί να ανακατευθύνει αρχεία σε κοινό αρχείο, κάτι που η εντολή more δεν μπορεί να κάνει. Με την εντολή

```
cat newer.program newest.program > all.programs
```

τα αρχεία newer.program και newest.program ανακατευθύνονται και συνενώνονται στο αρχείο all.programs.

- Εντολή mkdir: δημιουργεί νέους καταλόγους. Π.χ. με

```
mkdir newdirectory
```

Δημιουργείται ένας υποκατάλογος με το όνομα newdirectory στον τρέχοντα κατάλογο.

- Εντολή touch: δημιουργεί ένα κενό αρχείο, και αν συνοδεύεται με τη σημαία -t ενημερώνει την ημερομηνία και ώρα ενός αρχείου. Π.χ. με

```
touch file.create
```

Δημιουργείται στον τρέχοντα κατάλογο το κενό αρχείο file.create, ενώ με την εντολή

```
touch -t 12312359 old.file
```

αλλάζει η ημερομηνία δημιουργίας του αρχείου με την ημερομηνία 31 Δεκεμβρίου, ώρα 23 και 59 λεπτά.

- Εντολή `cp`: αντιγράφει ένα κατάλογο σε άλλο. Δέχεται τις σημαίες `-r` και `-i` και τα ονόματα του παλιού και του νέου καταλόγου. Η πρώτη εξασφαλίζει την αντιγραφή ενός ολόκληρου καταλόγου με τους υποκαταλόγους του, ενώ η δεύτερη προειδοποιεί πριν γίνει εγγραφή σε υπάρχον αρχείο.
- Εντολή `grep` (**G**lobal **R**egular **E**xpression **P**rint): Όταν δοθεί ένας κατάλογος αρχείων ή τυπικής εισόδου (standard input) προς είσοδο, το `grep` αναζητά γραμμές κειμένου που ταιριάζουν σε μια ή περισσότερες κανονικές εκφράσεις και δίνει σαν έξοδο μόνο τις γραμμές που ταιριάζουν.
- Εντολή `rmdir`: διαγράφει ένα κατάλογο. Για να διαγραφεί ένας κατάλογος πρέπει προηγουμένως να έχουν διαγραφεί όλοι οι υποκατάλογοι του και τα αρχεία του.
- Εντολή `find`: βρίσκει αρχείο που δεν είναι γνωστή η θέση του στο δένδρο των καταλόγων. Η αναζήτηση αρχίζει από τον τρέχοντα κατάλογο που δηλώνεται στην εντολή με τελεία. Η σύνταξη της εντολής έχει ως εξής:

```
find . -name_file -print
```

Με τη σημαία `-print` καθορίζεται η οθόνη ως το μέσο στο οποίο θα εμφανισθεί το αποτέλεσμα

Ανακατεύθυνση – Διοχέτευση

Η καθιερωμένη είσοδος στο UNIX είναι το πληκτρολόγιο, ενώ η καθιερωμένη έξοδος και η καθιερωμένη έξοδος λαθών είναι η οθόνη. Αυτά όμως μπορούν να αλλάξουν και να γίνει είσοδος και έξοδος από κάποιο αρχείο ή συσκευή (και οι συσκευές είναι αρχεία). Η είσοδος αλλάζει με τον χαρακτήρα `<` και η έξοδος με τον χαρακτήρα `>`. Αυτό λέγεται ανακατεύθυνση (redirection) ενώ υπάρχει και η δυνατότητα διοχέτευσης (piping) που στέλνει την έξοδο μιας εντολής σαν είσοδο σε μια άλλη εντολή. Αυτό γίνεται με το σύμβολο `|` ή με την εντολή `tee`. Η διοχέτευση ανακατευθύνει την έξοδο στο `stdin` για την νέα εντολή.

Π.χ.

```
cat file1 > file2
ps -ef | grep msav
mail msav < file1
```

Εντολή `xargs`

Η εντολή `xargs` επιτρέπει την κατασκευή command line arguments από την standard input (`stdin`). Έτσι, αν κάποια εντολή, όπως η `cp` και η `echo` δέχονται μόνο command line arguments και δεν διαβάζουν από `stdin`, τότε η `xargs` μας δίνει την δυνατότητα να τις καλέσουμε μέσω διοχέτευσης. Για παράδειγμα

```
ls | xargs echo
```

Το βοηθητικό πρόγραμμα `make`

Σκοπός του βοηθητικού προγράμματος (utility) `make` είναι να ορίσει αυτόματα ποια κομμάτια ενός μεγάλου προγράμματος χρειάζονται να ξανά μεταγλωττιστούν και ορίζει τις εντολές για αυτό το σκοπό. Τα προγράμματα μπορούν να είναι γραμμένα σε οποιαδήποτε γλώσσα της οποίας ο μεταγλωττιστής μπορεί να ξεκινήσει να εκτελείται με εντολές κελύφους. Επίσης, το utility `make` μπορεί να χρησιμοποιηθεί για να περιγράψει

οποιαδήποτε εργασία στην οποία κάποια αρχεία πρέπει να ενημερωθούν αυτόματα όταν κάποια άλλα αρχεία διαφοροποιηθούν. Για να χρησιμοποιηθεί το make, πρέπει να δημιουργηθεί ένα αρχείο που ονομάζεται Makefile το οποίο περιγράφει τις σχέσεις μεταξύ των αρχείων ενός προγράμματος και ορίζει τις εντολές με τις οποίες ενημερώνεται το κάθε αρχείο.

Παράδειγμα

episode_rename.c

```
#include "utilities.h"

void main(int argc, char *argv[]){

    if (argc != 2){
        /* Number of inputs should be 2; otherwise aborts */
        fprintf(stderr, "Wrong number of inputs! Aborting...\n");
        exit(EXIT_FAILURE);
    }

    char *filename = argv[1];    /* argv[0] is the executable name */
    fprintf(stdout, "\"%s\"", filename);
    fprintf(stdout, " ");

    char *new_filename = updateFilename( filename );
    fprintf(stdout, "%s\n", new_filename);
    free( new_filename );

    return;
}
```

utilities.c

```
#include "utilities.h"

char *updateFilename( char *filename ){

    char tmp1[20], tmp2[20];
    int season, episode;

    char *result = (char *)malloc(7*sizeof(char));
    sscanf( filename, "%s %d %s %d", tmp1, &season, tmp2, &episode );
    sprintf( result, "S%02d_E%02d", season, episode );

    return result;
}
```

utilities.h

```
#ifndef UTILITIES_H
#define UTILITIES_H
```

```
#include <stdio.h>
#include <stdlib.h>
```

```
char *updateFilename( char *filename );
```

```
#endif
```

Για να μεταγλωττίσετε τα δύο παραπάνω αρχεία, θα πρέπει να δώσετε την εξής εντολή στο προτρεπτικό κελύφους:

Για να χρησιμοποιήσετε το utility make, γράφετε πρώτα το makefile

```
CC = gcc
RM = rm -f

all: episode_rename

# final link for executable
episode_rename: episode_rename.o utilities.o
    $(CC) $^ -o $@

# generate objects
%.o: %.c
    $(CC) -c $<

# clean temporary files
clean:
    $(RM) *.o *~

# remove executable
purge: clean
    $(RM) episode_rename
```

και μετέπειτα δίνετε στο προτρεπτικό κελύφους τις εντολές

make

(Προσοχή: Στο makefile, μετά από κάθε κλήση της εντολής gcc, πρέπει να ακολουθεί κενή γραμμή. Επίσης το περιθώριο πριν τις εντολές gcc εισάγεται με το πλήκτρο TAB και όχι με κενά.)

Ποιό το αποτέλεσμα του προηγούμενου προγράμματος αν καλέσουμε την ακόλουθη εντολή;

```
ls | xargs -I {} -d '\n' episode_rename {} | mv -n2
```

Advanced Packaging Tool (APT)

Στο Unix μας δίνετε η δυνατότητα να διαχειριστούμε διάφορες εφαρμογές σαν λογισμικά πακέτα (packages) με το εργαλείο APT. Έτσι το APT προσφέρει μια διεπαφή χρήστη που χειρίζεται την εγκατάσταση και απεγκατάσταση λογισμικών. Απλοποιεί την διαδικασία χειρισμού λογισμικών σε Unix-like συστήματα υπολογιστών με το να συλλέγει, να συντονίζει και να εγκαθιστά λογισμικά πακέτα αυτόματα.

Για να χρησιμοποιήσουμε το εργαλείο APT πρέπει να έχουμε δικαιώματα υπερχρήστη ή δικαιώματα να χρησιμοποιήσουμε το APT σε ένα συγκεκριμένο λογισμικό πακέτο. Γι' αυτό τον λόγο πριν από κάθε εντολή κελύφους, εισάγουμε την εντολή sudo (superuser do). Η εντολή apt-get μπορεί να χρησιμοποιηθεί για να

εγκαταστήσουμε καινούρια λογισμικά πακέτα, να αναβαθμίσουμε ήδη εγκατεστημένα λογισμικά πακέτα και επίσης να αναβαθμίσουμε ολόκληρο το λειτουργικό σύστημα.

Π.χ. η εντολή

```
sudo apt-get install nmap
```

εγκαθιστά το nmap λογισμικό πακέτο.

Η εντολή

```
sudo apt-get remove nmap
```

αφαιρεί το nmap λογισμικό πακέτο.

Η εντολή apt-cache χειρίζεται και συλλέγει πληροφορίες για τα λογισμικά πακέτα που βρίσκονται στην κρυφή μνήμη του APT.

Π.χ. Η εντολή

```
apt-cache showpkg amsn
```

μας δίνει κάποιες γενικές πληροφορίες για το λογισμικό πακέτο amsn.