



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Τμήμα Πληροφορικής

ΕΠΛ371 - Προγραμματισμός Συστημάτων

ΑΣΚΗΣΗ 3 – Υλοποίηση προγράμματος για την προσπέλαση δεδομένων του συστήματος αρχείων (ls -ialR)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Υπεύθυνος εργαστηρίου: Πάυλος Αντωνίου

Ημερομηνία Ανάθεσης: Παρασκευή, 30/03/2018

Ημερομηνία Παράδοσης: Δευτέρα, 16/04/2018 και ώρα 15:00

(η λύση να υποβληθεί σε zip μέσω του Moodle)

<http://www.cs.ucy.ac.cy/courses/EPL371>

I. Στόχος Άσκησης

Στόχος αυτής της άσκησης είναι η εξοικείωση με προχωρημένες τεχνικές προγραμματισμού στο περιβάλλον UNIX. Συγκεκριμένα, σε αυτή την άσκηση θα έχετε την δυνατότητα να κάνετε πρακτική εξάσκηση των εννοιών που έχετε διδαχθεί στις διαλέξεις όπου είδαμε τη *Δομή Αρχείων*, το *Χαμηλού Επιπέδου I/O*, *Διαχείριση μέτα-πληροφοριών αρχείων* (<sys/stat.h>), *Διαχείριση αρχείων με κλήσεις συστήματος και Προσπέλαση και Επεξεργασία καταλόγων* (dirent.h) μέσω κλήσεων βιβλιοθήκης. Η υλοποίηση της άσκησης προϋποθέτει ότι έχετε κατανοήσει την δομή του συστήματος αρχείων στο UNIX καθώς επίσης την λειτουργία των I-nodes.

Η άσκηση περιλαμβάνει την υλοποίηση ενός υποσυνόλου λειτουργιών της εντολής ls, συγκεκριμένα των επιλογών (-ialR), χωρίς την κλήση της οικογένειας εντολών exec. Το πρόγραμμα σας θα πρέπει επίσης να παρέχει την δυνατότητα στους χρήστες να ορίσουν το μέγιστο βάθος διερεύνησης της αναδρομής (κάτι το οποίο δεν παρέχεται από την τυπική εντολή ls). Οι λειτουργίες του προγράμματος σας και το αναμενόμενο αποτέλεσμα περιγράφονται αναλυτικότερα στην συνέχεια.

III. Περιγραφή της εντολής ls (list segments)

Η εντολή ls παρουσιάζει το περιεχόμενο ενός καταλόγου που δίδεται σαν παράμετρος. Εάν ο χρήστης παραλείψει να δώσει παράμετρο τότε η εντολή παρουσιάζει αυτόματα το περιεχόμενο του τρέχοντος καταλόγου «.». Επίσης ο χρήστης έχει την δυνατότητα να δώσει σαν παράμετρο πολλαπλούς καταλόγους. Σε αυτή την περίπτωση, θα παρουσιάσουν τα περιεχόμενα όλων των καταλόγων που ορίζονται από το χρήστη. Παρόλο που η ls έχει μια πλειάδα επιλογών, σε αυτή την άσκηση θα μας απασχολήσουν οι πιο κάτω επιλογές καθώς επίσης κάποιες εντολές οι οποίες προσφέρονται αυτή τη στιγμή από την ls και οι οποίες θα περιγραφούν στη συνέχεια.

ls - list directory contents

- R list subdirectories recursively (using Depth First Search)
- l use a long listing format
- I print the index (inode) number of each file
- a include all entries (including hidden files starting with .)

Στην επόμενη εκτέλεση παρουσιάζουμε δυο παραδείγματα εκτέλεσης. Στόχος της άσκησης είναι να προσομοιώσετε αυτή την λειτουργία. Η εντολή `ls -ialR` επιστρέφει εγγραφές της μορφής:

```
<inode,permissions,links,owner,group,size,modification_date,name>
```

Το δικό σας πρόγραμμα θα πρέπει να τυπώνει τα ίδια ακριβώς αποτελέσματα που τυπώνει η εντολή `ls -ialR`. Ένα παράδειγμα εκτέλεσης της εντολής `ls -ialR` δίδεται πιο κάτω:

```
$ ls -ialR public_html/ep1003/
public_html/ep1003/:
total 1032
6671430604 drwxr-xr-x. 5 csp5pa1 tspecial 4096 Feb 27 16:39 .
4373503530 drwxr-xr-x. 9 csp5pa1 tspecial 4096 Mar 30 02:33 ..
6671430606 -rw-r--r--. 1 csp5pa1 tspecial 5839 Jan 29 12:47 announcements.html
6671430605 -rw-r--r--. 1 csp5pa1 tspecial 5817 Mar 19 13:01 assignments.html
6671430607 -rw-r--r--. 1 csp5pa1 tspecial 4895 Mar 12 13:21 contract.html
7225507548 -rw-r--r--. 1 csp5pa1 tspecial 232015 Mar 5 13:24 ep1003-spring2018.pdf
248851459 drwxr-xr-x. 2 csp5pa1 tspecial 40 Feb 12 01:18 ergasies
2160270148 drwxr-xr-x. 2 csp5pa1 tspecial 99 Jan 29 12:47 images
6671430608 -rw-r--r--. 1 csp5pa1 tspecial 8014 Mar 12 13:21 index.html
4380989636 drwxr-xr-x. 2 csp5pa1 tspecial 4096 Mar 26 13:27 labs
6671430609 -rw-r--r--. 1 csp5pa1 tspecial 10533 Mar 26 15:43 labs.html
6671430585 -rw-r--r--. 1 csp5pa1 tspecial 735797 Feb 26 13:18 midterm-EPL003.pdf
6671430610 -rw-r--r--. 1 csp5pa1 tspecial 5346 Jan 29 12:47 style.css
6671430613 -rw-r--r--. 1 csp5pa1 tspecial 18073 Feb 26 13:18 XMLMid.xlsx

public_html/ep1003/ergasies:
total 24
248851459 drwxr-xr-x. 2 csp5pa1 tspecial 40 Feb 12 01:18 .
6671430604 drwxr-xr-x. 5 csp5pa1 tspecial 4096 Feb 27 16:39 ..
375809153 -rw-r--r--. 1 csp5pa1 tspecial 20410 Feb 12 09:17 ExcelExcercise.xlsx

public_html/ep1003/images:
total 532
2160270148 drwxr-xr-x. 2 csp5pa1 tspecial 99 Jan 29 12:47 .
6671430604 drwxr-xr-x. 5 csp5pa1 tspecial 4096 Feb 27 16:39 ..
2215711643 -rw-r--r--. 1 csp5pa1 tspecial 33296 Jan 29 12:47 cslogo.png
2215711644 -rw-r--r--. 1 csp5pa1 tspecial 13312 Jan 29 12:47 Thumbs.db
2216466862 -rw-r--r--. 1 csp5pa1 tspecial 482415 Jan 29 12:47 UCY_Campus.pdf
2215793797 -rw-r--r--. 1 csp5pa1 tspecial 1711 Jan 29 12:47 UCY_logo.gif

public_html/ep1003/labs:
total 11592
4380989636 drwxr-xr-x. 2 csp5pa1 tspecial 4096 Mar 26 13:27 .
6671430604 drwxr-xr-x. 5 csp5pa1 tspecial 4096 Feb 27 16:39 ..
4380989637 -rw-r--r--. 1 csp5pa1 tspecial 466665 Jan 29 12:47 ExcelPracticel.pdf
4373579109 -rw-r--r--. 1 csp5pa1 tspecial 1788139 Mar 19 13:30 lab10.pdf
5207068911 -rw-r--r--. 1 csp5pa1 tspecial 742528 Mar 26 13:27 lab11.pdf
4380989638 -rw-r--r--. 1 csp5pa1 tspecial 2490945 Feb 1 09:21 lab3.pdf
4511667145 -rw-r--r--. 1 csp5pa1 tspecial 1589224 Feb 12 01:04 lab4.pdf
4624616062 -rw-r--r--. 1 csp5pa1 tspecial 1018991 Feb 12 13:29 lab5.pdf
5159510441 -rw-r--r--. 1 csp5pa1 tspecial 13689 Feb 12 11:18 lab5-problems.xlsx
4565590543 -rw-r--r--. 1 csp5pa1 tspecial 22868 Feb 15 23:37 lab5-solutions.xlsx
4636107406 -rw-r--r--. 1 csp5pa1 tspecial 1270202 Mar 5 13:01 lab8.pdf
4381127911 -rw-r--r--. 1 csp5pa1 tspecial 2428009 Mar 12 13:15 lab9.pdf
```

Εάν δεν δοθεί κάποια παράμετρος τότε η εντολή εκτελείται πάνω στο παρόν κατάλογο “.”:

```
$ls -ialR
.:
total 580
6483572892 drwxr-xr-x. 2 csp5pa1 tspecial 4096 Apr 20 2017 .
4373503530 drwxr-xr-x. 9 csp5pa1 tspecial 4096 Mar 30 02:33 ..
6483572894 -rw-r--r--. 1 csp5pa1 tspecial 117248 Apr 20 2017 anyplace-directions.jpg
6483572893 -rw-r--r--. 1 csp5pa1 tspecial 1756 Apr 20 2017 arrow78.png
6483572895 -rw-r--r--. 1 csp5pa1 tspecial 1395 Apr 20 2017 ccby.png
6483658450 -rw-r--r--. 1 csp5pa1 tspecial 4719 Apr 20 2017 close.png
```

```

6483658451 -rw-r--r--. 1 csp5pa1 tspecial 18861 Apr 20 2017 csucy.png
6483658471 -rw-r--r--. 1 csp5pa1 tspecial 1419 Apr 20 2017 office.gif
6483662390 -rw-r--r--. 1 csp5pa1 tspecial 99497 Apr 20 2017 pavlosantoniou.png
6483662402 -rw-r--r--. 1 csp5pa1 tspecial 388 Apr 20 2017 pdf.gif
6483662403 -rw-r--r--. 1 csp5pa1 tspecial 10057 Apr 20 2017 ucy.jpeg
7238263419 lrwxrwxrwx. 1 csp5pa1 tspecial 8 Mar 30 02:41 ucy.jpg -> ucy.jpeg
6483662405 -rw-r--r--. 1 csp5pa1 tspecial 281743 Apr 20 2017 zeinalipour.pdf
6483662404 -rw-r--r--. 1 csp5pa1 tspecial 8932 Apr 20 2017 zeinalipour-qr.gif
6522414701 -rw-r--r--. 1 csp5pa1 tspecial 10163 Apr 20 2017 zeinalipour.small.jpg

```

Σημειώστε ότι γίνεται χρήση της Κατά-Βάθος Διερεύνησης (Depth-First Traversal). Επίσης θα λαμβάνονται υπόψη στην παρουσίαση (μπροστά από τα permissions, βλέπε 2^η στήλη πιο πάνω) όλοι οι τύποι αρχείων είτε πρόκειται για regular αρχείο (-) ή για αρχείο (d) ή για συμβολικό σύνδεσμο (l).

Όπως προσέξατε από τα πιο πάνω παραδείγματα, όταν χρησιμοποιηθεί η παράμετρος -l τότε εμφανίζεται η λέξη total ακολουθούμενη από ένα αριθμό. Προτού αναλύσουμε πως προκύπτει ο αριθμός αυτός πρέπει να δώσουμε κάποιες επιπλέον πληροφορίες. Κάθε αρχείο και κάθε κατάλογος δεσμεύουν στο δίσκο ένα αριθμό από blocks για να αποθηκεύσουν τα περιεχόμενα τους. Για να δούμε πόσα blocks δεσμεύει κάθε αρχείο και κάθε κατάλογος στον τρέχοντα κατάλογο μπορούμε να εκτελέσουμε την εντολή ls -s και θα πάρουμε τα πιο κάτω αποτελέσματα:

```

8 as1                8 main.c
8 as2                12 moss.pl
4 as3                8 putty-0.62
4 as4                16 rambinsearch
12 as5               4 rambinsearch.c
4 bash.sh            4 shell.sh
4 binsearch.sh       204 stdlib.jar
4 bitcoin.conf       4840 words.txt
4 bitcoin.sh

```

Το κάθε block μπορεί να αποτελείται από ένα αριθμό από bytes πολλαπλάσιο του 512 bytes. Ο αριθμός 512 θεωρείται η ελάχιστη χωρητικότητα αποθήκευσης καθώς αντιστοιχεί στους τομείς (sectors) των πρώτων δίσκων που κυκλοφόρησαν. Στο πιο πάνω παράδειγμα το block size που χρησιμοποιείται από την ls είναι το 1024 bytes και αυτό μπορεί να διαπιστωθεί αν τρέξω την ίδια εντολή με το -h option που παρουσιάζει τα αποτελέσματα σε human readable μορφή. Έτσι η ls -sh δίνει:

```

8.0K as1             8.0K main.c
8.0K as2             12K moss.pl
4.0K as3             8.0K putty-0.62
4.0K as4             16K rambinsearch
12K as5              4.0K rambinsearch.c
4.0K bash.sh         4.0K shell.sh
4.0K binsearch.sh    204K stdlib.jar
4.0K bitcoin.conf    4.8M words.txt
4.0K bitcoin.sh

```

Το συνολικό άθροισμα των blocks (των 1024 bytes το καθένα) που χρησιμοποιούνται από τα περιεχόμενα του τρέχοντος καταλόγου είναι αυτό που φαίνεται δίπλα από τη λέξη total όταν τρέξουμε την εντολή ls -l :

```

total 5148
drwxr-sr-x. 52 csp5pa1 tspecial 4096 Nov 10 12:17 as1
drwxr-sr-x. 5 csp5pa1 tspecial 4096 Dec 6 09:11 as2
drwxr-sr-x. 7 csp5pa1 tspecial 4096 Nov 28 13:11 as3
drwxr-sr-x. 24 csp5pa1 tspecial 4096 Dec 6 09:10 as4
drwxr-sr-x. 4 csp5pa1 tspecial 8192 Dec 8 00:27 as5
-rwxrwxrwx. 1 csp5pa1 tspecial 171 Feb 25 2015 bash.sh
-rwxr-xr-x. 1 csp5pa1 tspecial 1003 Mar 30 00:17 binsearch.sh
-rw-r--r--. 1 csp5pa1 tspecial 1659 Feb 17 12:01 bitcoin.conf

```

```

-rwxr--r--. 1 csp5pa1 tspecial      315 Feb 17 12:10 bitcoin.sh
-rw-r--r--. 1 csp5pa1 tspecial     6989 Jan 26 18:16 main.c
-rwxr-xr-x. 1 csp5pa1 tspecial    10756 Mar 18 2016 moss.pl
drwxr-xr-x. 10 csp5pa1 tspecial     4096 May  4 2015 putty-0.62
-rwxr-xr-x. 1 csp5pa1 tspecial    13240 Mar 30 00:14 rambinsearch
-rw-r--r--. 1 csp5pa1 tspecial     1798 Mar 30 00:14 rambinsearch.c
-rwxr-xr-x. 1 csp5pa1 tspecial      100 Jan 30 10:40 shell.sh
-rw-r--r--. 1 csp5pa1 tspecial   207354 Aug 28 2017 stdlib.jar
-rw-r--r--. 1 csp5pa1 tspecial  4953680 Mar 29 10:34 words.txt

```

Ο αριθμός των blocks κάθε αρχείου (ή καταλόγου) ξεχωριστά μπορεί να υπολογιστεί και από την εντολή κελύφους stat. Αν τρέξω την εντολή stat words.txt θα λάβω τα πιο κάτω:

```

File: \u2018words.txt\u2019
Size: 4953680      Blocks: 9680      IO Block: 1048576 regular file
Device: 32h/50d   Inode: 2959340741  Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1220/ csp5pa1)   Gid: ( 242/tspecial)
Context: system_u:object_r:nfs_t:s0
Access: 2018-03-29 10:34:34.367586414 +0300
Modify: 2018-03-29 10:34:21.698674485 +0300
Change: 2018-03-29 10:34:28.964623988 +0300
Birth: -

```

Όπως φαίνεται πιο πάνω, ο αριθμός των blocks που δίνεται από τη stat είναι διπλάσιος από αυτόν της ls και ο λόγος είναι ότι η stat θεωρεί για τον υπολογισμό της ότι το κάθε block έχει μέγεθος 512 bytes (εξ' ου και υπολογίζει διπλάσια blocks). Αυτό επιβεβαιώνεται αν τρέξω την ls με block-size=512, χρησιμοποιώντας την εντολή ls --block-size=512 -s:

```

16 as1              16 main.c
16 as2              24 moss.pl
 8 as3              16 putty-0.62
 8 as4              32 rambinsearch
24 as5              8 rambinsearch.c
 8 bash.sh          8 shell.sh
 8 binsearch.sh     408 stdlib.jar
 8 bitcoin.conf     9680 words.txt
 8 bitcoin.sh

```

Στην υλοποίηση της δική σας ls -ialR να παρουσιάζετε τον αριθμό των blocks δίπλα από τη λέξη total με την ίδια λογική που τα παρουσιάζει η ls.

III. Βοηθητικές Συναρτήσεις

Για την διεκπεραίωση αυτής της άσκησης θα χρειαστείτε κάποιες συναρτήσεις τις οποίες δεν είδαμε στο μάθημα. Συγκεκριμένα θα χρειαστείτε τις συναρτήσεις (Κεφάλαιο 6 – Stevens & Rago):

```

#include <pwd.h>
struct passwd *getpwuid(uid_t uid);
Returns: Pointer to struct passwd if OK, NULL on error

```

```

#include <grp.h>
struct group *getgrgid(gid_t gid);
Returns: Pointer to struct group if OK, NULL on error

```

α) struct passwd *getpwuid(uid_t uid);

Αντιστοιχεί το uid το οποίο περιέχετε στο i-node ενός αρχείου σε μια πιο εύληπτη μορφή (π.χ. αντί 1240 δίδει dzeina). Η εντολή αυτή ουσιαστικά ψάχνει μέσα στο /etc/passwd αρχείο. Η struct passwd περιέχει το πεδίο pw_name το οποίο είναι τύπου (char *) και το οποίο περιέχει το όνομα του χρήστη.

β) struct group *getgrgid(gid_t gid);

Αντιστοιχεί το gid το οποίο περιέχετε στο i-node ενός αρχείου σε μια πιο εύληπτη μορφή (π.χ. αντί 243 δίδει visiting). Η εντολή αυτή ουσιαστικά ψάχνει μέσα στο /etc/group αρχείο. Η struct group περιέχει το πεδίο gr_name το οποίο είναι τύπου (char *) και το οποίο περιέχει το όνομα του χρήστη.

γ) Μελετήσετε το “man stat” για να κατανοήσετε όλες τις σταθερές της βιβλιοθήκης <sys/stat.h>

IV. Ζητούμενα Άσκησης

Πιο κάτω περιγράφονται οι διαφορές επιλογές τις οποίες πρέπει να διαχειρίζεται το πρόγραμμα σας:

\$ myls: Εκτελεί αντίστοιχη λειτουργία με την εντολή **ls -ialR**

\$ myls a/ b/ c/ : Εκτελεί αντίστοιχη λειτουργία με την εντολή **ls -ialR a/ b/ c/**

\$ myls -depth 2: Εκτελεί αντίστοιχη λειτουργία με την εντολή **ls -ialR** αλλά το βάθος διερεύνησης περιορίζεται σε δυο επίπεδα.

\$ myls -follow-symbolic: ο αλγόριθμος ακολουθά και τα συμβολικά αρχεία και καταλόγους. Εάν είναι κατάλογος τότε εκτυπώνει κανονικά και τα περιεχόμενα του. Σε αυτή την περίπτωση πρέπει ο αλγόριθμος σας να ελέγχει εάν ο κατάλογος έχει ήδη εκτυπωθεί έτσι ώστε να μην τον ξανά-εκτυπώσει (και ούτε να τον επεξεργαστεί περαιτέρω). Η έλεγχος αυτός πρέπει να είναι όσο το δυνατό πιο αποδοτικός γίνεται (θα θέλαμε δηλαδή να αποφύγουμε να σαρώσουμε ολόκληρη τη λίστα των προηγούμενων μονοπατιών). Σημειώστε ότι ο αλγόριθμος διερεύνησης σε όλες τις περιπτώσεις είναι Depth-First-Search.

\$ myls -output traversal-log.txt: εκτυπώνει σε αρχείο με όνομα **traversal-log.txt** το depth-first-search μονοπάτι το οποίο ακολουθήθηκε από την εντολή myls με παρόμοιο τρόπο με το παράδειγμα που ακολουθεί

```
./
./myslides/
./myslides/epl371/
./myslides/epl371/slide1.ppt
./myslides/epl371/slide2.ppt
./myslides/epl032/
./myslides/epl032/intro.ppt
./myslides/epl001/
./exc/
./exc2/
./exc2/as2.doc
./myfile.txt
```

Σημειώστε ότι η εφαρμογή σας πρέπει να υποστηρίζει οποιουδήποτε συνδυασμούς των πιο πάνω ορισμάτων και να παράγει το αναμενόμενο αποτέλεσμα.

V. Γενικοί Κανόνες

1. Το πρόγραμμα σας πρέπει να χρησιμοποιεί τεχνικές δομημένου προγραμματισμού με την χρήση συναρτήσεων, πολλαπλών αρχείων για καλύτερη δομή του πηγαίου κώδικα, Makefile, διαχείριση λαθών συστήματος με την perror κτλ.;

2. Για την επεξεργασία των επιλογών του προγράμματος σας θα ήταν καλό να κάνετε χρήση της βιβλιοθήκης getopt (για περισσότερες πληροφορίες “man 3 getopt”, όπου 3 δηλώνει “C library routines”)
3. Το πρόγραμμα σας πρέπει να ελαχιστοποιεί την χρήση πόρων του συστήματος (μνήμης, επεξεργαστή, στοίβας προγράμματος, κτλ);
4. Το πρόγραμμα πρέπει να μειώνει όσο το δυνατό περισσότερο τον χρόνο διεκπεραίωσης της εντολής.
5. Συμβουλευτείτε τις οδηγίες του ΕΠΛ232 για συγγραφή των σχετικών ελέγχων μονάδων, διάσπαση του προγράμματος σε πολλαπλά αρχεία, τεκμηρίωση, makefile, κτλ.

Καλή Επιτυχία !