

Εργαστηριακές ασκήσεις Βιοπληροφορικής.

Άσκηση 2^η «Εισαγωγή στο UNIX, μέρος δεύτερο»

Όνοματεπώνυμο :

Αριθμ. Μητρώου :

Σκοπός άσκησης :

- Περαιτέρω εξοικείωση με τις εντολές **cd**, **pwd**, **ls**, **mkdir**, **rmdir**, **cp**, **mv**, **rm**, **cat**, **more** (σελίδες 9-12 του εγχειριδίου).
- Σημασία και χρήση του συμβόλου **~**
- Σημασία και χρήση των χαρακτήρων υποκατάστασης ***** και **?**
- Καινούργια εντολή : **chmod** (σελίδες 25 και 26 του εγχειριδίου).
- Χρήση του κειμενογράφου για την τροποποίηση ενός εκ των αρχείων που είχατε δημιουργήσει στην 1^η άσκηση.

Χρησιμοποιώντας το εγχειρίδιο οδηγιών που σας δόθηκε, και με τη βοήθεια των διδασκόντων, απαντήστε τα παρακάτω ερωτήματα (για τις απαντήσεις σας χρησιμοποιήστε τις σελίδες αυτές οι οποίες θα πρέπει να επιστρέφονται στο τέλος της άσκησης).

Δίπλα από κάθε μια από τις εντολές που ακολουθούν γράψτε μια σύντομη και περιεκτική περιγραφή του τι επιτυγχάνουν :

cd

pwd

cd practicals

cd 1/

cd ../../Desktop/

pwd

cd

```
pwd
mkdir tmp/
cp practicals/1/function.c tmp/
cp practicals/1/main.c tmp/
cd tmp
ls -lF
cat main.c function.c
cat main.c function.c > test.c
more test.c
mv test.c 2del.c
ls -lF
rm -i 2del.c main.c function.c
cd ../
rmdir tmp/
cd practicals/1/
ls -lF
cd ../
mkdir 2/
cd 2/
cp ../1/program.c ./
ls -lF
cd
mkdir tmp/
```

Οι ακόλουθες δυο εντολές είναι ισοδύναμες ή όχι, και εάν όχι ποια η διαφορά τους ;

```
cp practicals/1/function.c tmp/
cp practicals/1/function.c /tmp/
```

Χρησιμοποιήστε τις επόμενες εντολές και από την έξοδο που παράγουν προσπαθήστε να προσδιορίσετε ποιο είναι το νόημα του συμβόλου ~ (σε τι αντιστοιχεί, ή αλλιώς, με τι είναι ισοδύναμο). Αφού προσδιορίσετε το νόημα του συμβόλου ~, γράψτε μια συνοπτική περιγραφή του τι επιτυγχάνετε με κάθε μια από αυτές τις εντολές.

```
cd ~
pwd
cd ~/practicals
pwd
cd /tmp
pwd
ls -lF
mv function.c ~/tmp/
ls -lF
cd ~/tmp/
pwd
rm function.c
cd ~
rmdir tmp/
```

Χρησιμοποιήστε τις επόμενες εντολές και από την έξοδο που παράγουν προσπαθήστε να προσδιορίσετε ποιο είναι το νόημα των συμβόλων * και ?. Αφού προσδιορίσετε το νόημα τους, περιγράψτε συνοπτικά τι επιδιώκουμε να επιτύχουμε με τη χρήση αυτών των εντολών. Την παράμετρο **-d** που χρησιμοποιούμε με την εντολή **ls** αγνοήστε την (ή χρησιμοποιήστε την εντολή **man ls** για να μάθετε τη σημασία της)

```
cd ~
cd ../
ls
ls -d b10000*
ls -d b10001*
ls -d b1*4
ls -d *4
ls -d *03*
ls -d b100?01
ls -d b1000?3
cd
ls -d .*
ls -d .desk*
```

Χρησιμοποιώντας τις πληροφορίες που δίδονται στις σελίδες 25 και 26 του εγχειριδίου, προσδιορίστε τι επιτυγχάνουν οι κάτωθι εντολές. Παρατηρήστε πως αλλάζει η έξοδος από τις εντολές **ls** καθώς αλλάζετε την προστασία των αρχείων σας.

```
cd
cd practicals/1/
ls -lF
chmod go+r *
ls -lF
cd ../2/
chmod go+r *
cd ..
ls -lF
chmod go+rx 1/ 2/
ls -lF
cd ..
ls -lF
chmod go+rx practicals/
ls -lF
cd ..
ls -dlF ~
chmod go+rx ~
ls -dlF ~
```

Χρησιμοποιώντας το τερματικό κάποιου συναδέλφου σας επιβεβαιώστε ότι είναι εφικτό κάποιος άλλος χρήστης (αυτός από τον υπολογιστή του οποίου κάνετε τη δοκιμή) να διαβάσει τα αρχεία που βρίσκονται στον υποκατάλογο **practicals/1/ του δικού σας home directory** (χρησιμοποιήστε τις εντολές **cat** ή **more** για να σιγουρευτείτε ότι όντως μπορείτε να διαβάσετε τα αρχεία **main.c, function.c, program.c**, κλπ.). Αφού επιβεβαιώσετε ότι όντως τα αρχεία είναι προσβάσιμα από τους άλλους χρήστες, μπορείτε (εάν μοιράζετε τον υπολογιστή με κάποιον άλλο), να κάνετε logout και στη συνέχεια να κάνει login ο συνάδελφος με τον οποίο μοιράζετε τον υπολογιστή, ο οποίος θα πρέπει να πια σε θέση να αντιγράψει (με την εντολή **cp**) τα αρχεία που είχατε δημιουργήσει κατά την διάρκεια της προηγούμενης άσκησης. Η αντιγραφή των αρχείων στην περιοχή του δεύτερου χρήστη είναι προαιρετική σε αυτό το στάδιο (σε μια από τις επόμενες ασκήσεις θα χρησιμοποιήσουμε την εντολή **tar** για να μεταφέρουμε ολόκληρα σύνολα καταλόγων, υποκαταλόγων και απλών αρχείων).

Χρησιμοποιήστε τον κειμενογράφο της επιλογής σας για να τροποποιήσετε το αρχείο **program.c** το οποίο βρίσκεται στον υποκατάλογο **practicals/2/** του κορυφαίου σας καταλόγου. Η τροποποίηση θα πρέπει να είναι τέτοια ώστε το αρχείο να γίνει όπως φαίνεται παρακάτω

```
#include <gl.h>
#include <stdio.h>
#include <math.h>

float function();
float p1, p2, p3;

main()
{
    long        winid;
    long        originX, originY;
    Screencoord left, right, bottom, top;
    float       x;
    float       trans;

    printf("Enter parameters [3] : \n");
    scanf("%f %f %f", &p1, &p2, &p3);
    foreground();
    prefsiz( 800, 800 );
    prefposition( 50, 850, 50, 850 );
    winid = winopen( "Practical 2" );
    winset ( winid );

    getviewport(&left, &right, &bottom, &top);
    getorigin( &originX, &originY );
    color(0);
    clear();
    viewport( left, right, bottom, top );

    color(1);
    for ( trans=0 ; trans < 2000 ; trans += 0.200 )
    {
        int k;
        k = (int)(trans) % 48;
        if ( k < 24 )
            color( 32 + k );
        else
            color( 55 + 24 - k );
        move2( 1, function( 1, trans));
        for ( x=1 ; x <= 800 ; x += 1.0 )
            draw2( x, function(x, trans) );
    }
}

float function(x, trans)
float x;
float trans;
{
    float y;
    y = 248.0*(sin((x+trans)/p1)/p2)*sin((x+trans)/p3)/(x/trans)+250.0 ;
    return( y );
}
```

Ελέγξτε ότι το αρχείο πληκτρολογήθηκε σωστά με την εντολή¹

```
/usr/freeware/bin/gcc -O2 -o graph program.c -lgl -lm
```

Εάν όλα έχουν πάει καλά, η παραπάνω εντολή δεν θα πρέπει να παράγει έξοδο, ενώ ένα αρχείο με το όνομα **graph** θα πρέπει να έχει δημιουργηθεί στον τρέχοντα κατάλογο (ελέγξτε ότι όντως δημιουργήθηκε χρησιμοποιώντας την εντολή **ls**). Δώστε την εντολή

```
./graph
```

για να εκτελέσετε («τρέξετε») το πρόγραμμα που δημιουργήσατε. Το πρόγραμμα περιμένει από εσάς να πληκτρολογήσετε αριθμητικές τιμές για τρεις παραμέτρους. Δοκιμάστε να πληκτρολογήσετε

```
1.0 10.0 40.0
```

(ακολουθούμενο από ENTER). Εάν δεν υπάρχουν τυπογραφικά σφάλματα στο αρχείο **program.c** ένα νέο παράθυρο θα δημιουργηθεί στο οποίο θα δείτε την γραφική παράσταση μιας κάπως πολύπλοκης συνάρτησης. Δοκιμάστε να ξανατρέξετε το πρόγραμμα, αυτή τη φορά με παραμέτρους

```
80.0 1.0 60.0
```

¹ Η εντολή αυτή δεν πραγματοποιεί ορθογραφικό έλεγχο του αρχείου που δημιουργήσατε. Περιλήπτικά αυτό που συμβαίνει είναι το εξής : Το αρχείο που ετοιμάσατε (το **program.c**) περιέχει ένα πρόγραμμα υπολογιστών γραμμένο στη γλώσσα προγραμματισμού C. Αυτό το πρόγραμμα είναι γραμμένο σε μια γλώσσα «υψηλού» επιπέδου, δηλαδή σε μια γλώσσα που είναι πλησιέστερη στη μορφή επικοινωνίας που χρησιμοποιείται από το είδος μας, απ'ότι σε μια γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο των εργασιών ηλεκτρονικών κυκλωμάτων (που είναι ο υπολογιστής). Για αυτό το λόγο είναι αναγκαία η μετάφραση του αρχικού προγράμματος (σε C), σε μια άλλη γλώσσα τόσο χαμηλού επιπέδου ώστε να είναι δυνατή η εκτέλεση των απαιτούμενων υπολογισμών από τα κυκλώματα του υπολογιστή σας. Η γλώσσα αυτή χαμηλού επιπέδου είναι γνωστή ως «κώδικας μηχανής» και είναι εντελώς ακατάληπτη για την τεράστια πλειοψηφία των ανθρώπων (μπορείτε να πάρετε μια ιδέα του τι εννοείται με το «ακατάληπτη», δίνοντας την εντολή `vi graph`). Η εντολή `/usr/freeware/bin/gcc` εκτελεί («τρέχει») ένα πρόγραμμα (με το όνομα **gcc**, το οποίο βρίσκεται στον κατάλογο `/usr/freeware/bin`) το οποίο μεταφράζει το αρχικό πρόγραμμα (σε C, στο αρχείο **program.c**) σε ένα πρόγραμμα σε κώδικα μηχανής (το αρχείο με το όνομα **graph**). Το αρχείο **graph** περιέχει ένα άμεσα εκτελέσιμο πρόγραμμα (τουλάχιστον για τον συγκεκριμένο υπολογιστή στον οποίο δημιουργήθηκε), και θα μπορούσε κάλλιστα να γίνει μια εντολή του συστήματος με τον ίδιο τρόπο που είναι οι εντολές **ls**, **pwd**, **cat**, κλπ.