

I. ΔΙΕΡΓΑΣΙΕΣ

Δημιουργείτε ένα πρόγραμμα τύπου “launcher”, το οποίο διαθέτει ένα μενού από έτοιμες εφαρμογές και ο χρήστης επιλέγει ποια θα εκτελέσει κάθε φορά. Για παράδειγμα, μπορεί να έχει το παρακάτω μενού:

1. run firefox
2. run firefox (and wait)
3. run gedit
4. run gedit (and wait)
5. run gnome-calculator
6. run gnome-calculator (and wait)
0. exit

(α) Λειτουργία:

Για την εκτέλεση κάθε εφαρμογής θα πρέπει να δημιουργείτε μία διεργασία-παιδί (με `fork()`), η οποία στη συνέχεια, με χρήση της `exec1()`, εκτελεί τη ζητούμενη εφαρμογή. Με τις επιλογές 2, 4 και 6, ο χρήστης δηλώνει ότι θέλει να περιμένει να ολοκληρωθεί η εφαρμογή και μετά να συνεχίσει, ενώ με τις επιλογές 1, 3 και 5 ο γονέας είναι ελεύθερος να συνεχίσει άμεσα και ανεξάρτητα.

(β) Δομή:

Το μενού θα πρέπει να υλοποιηθεί ως ένας πίνακας από 7 δομές (`struct`). Η κάθε δομή, θα πρέπει να περιέχει

- το λεκτικό της κάθε επιλογής (π.χ. “2. run firefox (and wait)”)
- την διαδρομή του αντίστοιχου προγράμματος (π.χ. “/usr/bin/firefox”)
- το όνομα του αντίστοιχου προγράμματος (π.χ. “firefox”)

ώστε τα πεδία της κάθε δομής να χρησιμοποιούνται τόσο για την εκτύπωση του μενού όσο και για τα ορίσματα που δίνονται στην `exec1()`. Η πλήρης διαδρομή μίας εφαρμογής μπορεί να βρεθεί με την εντολή τερματικού `which`.

II. ΔΙΕΡΓΑΣΙΕΣ ΚΑΙ ΑΠΛΟΙ ΑΓΩΓΟΙ

Σχεδιάστε ένα πρόγραμμα το οποίο:

- (α) Δημιουργεί μία θυγατρική διεργασία με την οποία θα επικοινωνεί μέσω ενός αγωγού.
- (β) Η **θυγατρική διεργασία** διαβάζει ακεραίους αριθμούς από τον χρήστη, και τους περνάει στη γονική διεργασία μέσω του αγωγού.
- (γ) Η γονική διεργασία ελέγχει τους αριθμούς που λαμβάνει και τους θετικούς από αυτούς τους γράφει σε ένα αρχείο `κειμένου positive.txt`.

Το πρόγραμμα τελειώνει όταν ο χρήστης δώσει τον αριθμό `-1`. Φροντίστε να τερματίσουν όλα σωστά και ομαλά, και να ελέγξετε αν αποτυπώθηκαν όλα όσα πρέπει στο αρχείο `positive.txt`.

III. ΔΙΕΡΓΑΣΙΕΣ ΚΑΙ ΠΡΟΧΩΡΗΜΕΝΟΙ ΑΓΩΓΟΙ

Το πρόγραμμα `gnuplot` είναι μία από τις δημοφιλέστερες εφαρμογές για σχεδίαση γραφικών παραστάσεων. Εκτελώντας στο τερματικό σας:

```
$ gnuplot
```

Ξεκινά η εφαρμογή (εμφανίζεται το prompt `gnuplot>`), η οποία περιμένει απλές εντολές από το πληκτρολόγιο για να κάνει τη δουλειά που θέλετε. Για παράδειγμα, μπορείτε να δώσετε:

```
gnuplot> plot sin(x);
```

προκειμένου να σας ζωγραφίσει (στην οθόνη) τη γραφική παράσταση του ημιτόνου. Ή μπορείτε να δώσετε:

```
gnuplot> help plot;
```

για να λάβετε βοήθεια στο πώς σχεδιάζεται γραφικές παραστάσεις. Η εφαρμογή τερματίζει με την εντολή exit:

```
gnuplot> exit;  
$
```

Από εσάς ζητείται να υλοποιήσετε ένα πρόγραμμα το οποίο εκτελεί την εφαρμογή gnuplot προκειμένου να σχεδιάσει μία γραφική παράσταση της αρεσκείας σας. Πιο συγκεκριμένα:

- α) Το πρόγραμμά σας θα πρέπει να δημιουργεί μία διεργασία-παιδί και έναν αγωγό (pipe) προκειμένου να επικοινωνεί μαζί της.
- β) Η διεργασία-παιδί θα πρέπει μέσω `execl()` να εκτελέσει την gnuplot και να της δίνει στην standard είσοδό της ότι λαμβάνει από τον αγωγό του γονέα (με χρήση συναρτήσεων τύπου `dup`).
- γ) Ο γονέας θα πρέπει να γράφει στον αγωγό τις εντολές προς το gnuplot. Στο τέλος θα πρέπει να περιμένει είσοδο από τον χρήστη και θα στέλνει την εντολή `exit` για να τερματίσει το gnuplot.

IV. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ ΜΕ ΔΙΕΡΓΑΣΙΕΣ

Σχεδιάστε ένα πρόγραμμα το οποίο ζητά από τον χρήστη έναν θετικό αριθμό N και στη συνέχεια δημιουργεί θυγατρική διεργασία ώστε γονέας και παιδί να μοιραστούν τον υπολογισμό του αθροίσματος $1 + 2 + \dots + N$. Δηλαδή:

- (α) Ο γονέας υπολογίζει το άθροισμα από 1 μέχρι το $N/2$.
- (β) Το παιδί υπολογίζει το άθροισμα από $N/2 + 1$ έως N .
- (γ) Το παιδί στέλνει (με όποιον τρόπο θέλετε) το αποτέλεσμα στο γονέα.
- (δ) Στο τέλος, ο γονέας αθροίζει με το δικό του αποτέλεσμα και εκτυπώνει στην οθόνη.

Συγχαρητήρια, μόλις κάνατε τον πρώτο σας “παράλληλο” υπολογισμό ;-)

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ ΜΕ ΑΓΩΓΟΥΣ

Σχεδιάστε ένα πρόγραμμα το οποίο:

- (α) Δημιουργεί μία θυγατρική διεργασία με την οποία θα επικοινωνεί μέσω **δύο (2)** αγωγών—στον έναν θα γράφει ο γονέας και στον άλλον θα γράφει το παιδί.
- (β) Η γονική διεργασία διαβάζει ακεραίους αριθμούς από τον χρήστη, και τους περνάει στη θυγατρική διεργασία μέσω του πρώτου αγωγού, μέχρι να δοθεί το -1 .
- (γ) Η θυγατρική διεργασία λαμβάνει τους αριθμούς από τον γονέα και θα πρέπει να υπολογίσει το *άθροισμα και το μέσο όρο τους* (ο τελευταίος θα είναι `double`).
- (δ) Στο τέλος, η θυγατρική διεργασία θα στείλει, μέσω του *δεύτερου* αγωγού, το άθροισμα και τον μέσο όρο των αριθμών στην γονική διεργασία. Η τελευταία θα τα εκτυπώσει και θα τερματίσει.

VI. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ ΜΕ ΠΡΟΧΩΡΗΜΕΝΟΥΣ ΑΓΩΓΟΥΣ

Προσπαθήστε να ελέγξετε κι άλλες εφαρμογές, όπως στην Άσκηση III. Για παράδειγμα, μπορείτε να το κάνετε για την εφαρμογή `matlab` ή `octave`, εφόσον τις γνωρίζετε.