

## Εργασία 4: Περιβάλλον ελεγχόμενης εκτέλεσης προγραμμάτων μέσω σημάτων

Υλοποιήστε ένα «περιβάλλον» εκτέλεσης προγραμμάτων το οποίο να υποστηρίζει τις εξής εντολές χρήστη:

Εντολή	Ορίσματα	Περιγραφή
exec	<progname> <args>	Εκτέλεση του προγράμματος progname με ορίσματα args, μέσω ξεχωριστής διεργασίας (fork-exec).
term	<pid>	Αποστολή σήματος SIGTERM στη διεργασία με αναγνωριστικό pid.
sig	<pid>	Αποστολή σήματος SIGUSR1 στη διεργασία με αναγνωριστικό pid.
list		Πληροφορίες για τις διεργασίες που εξακολουθούν να υφίστανται: αναγνωριστικό, όνομα προγράμματος, ορίσματα, κατάσταση (αν τρέχει).
quit		Τερματισμός περιβάλλοντος εκτέλεσης.

Με την εντολή `exec` αρχίζει η εκτέλεση ενός προγράμματος μέσω μιας νέας διεργασίας που δημιουργεί το περιβάλλον για αυτό τον σκοπό μέσω `fork-exec`. Τα αναγνωριστικά των διεργασιών πρέπει να διατηρούνται σε ένα πίνακα ή λίστα που ανανεώνεται όταν δημιουργείται / τερματίζεται μια διεργασία (σημείωση: μια διεργασία μπορεί να τερματίσει μόνη της ή λόγω εξωτερικών παραγόντων, χωρίς ο χρήστης να δώσει την εντολή `term`).

Τα προγράμματα που ζητά ο χρήστης να εκτελεστούν μέσα από το περιβάλλον εκτέλεσης μπορεί να είναι οτιδήποτε (για πρακτικούς λόγους καλό είναι να μην διαβάζουν από την συμβατική είσοδο και να μην εκτυπώνουν πολλή πληροφορία στην συμβατική έξοδο τους). Επίσης, θα χρειαστεί να υλοποιήσετε εσείς ένα πρόγραμμα που ικανοποιεί συγκεκριμένες προδιαγραφές (βλ. επόμενη σελίδα).

Πριν την εκκίνηση μιας διεργασίας που εκτελεί πρόγραμμα που ζήτησε ο χρήστης, το περιβάλλον εκτέλεσης πρέπει να αποκαθιστά το συμβατικό χειρισμό του σήματος SIGUSR1 γι αυτή τη διεργασία. Επίσης, το ίδιο το περιβάλλον χειρίζεται το SIGUSR1 προωθώντας το σε όσες διεργασίες-παιδιά είναι ακόμα ενεργές (δεν έχουν ήδη τερματιστεί).

Με την εντολή `quit`, το περιβάλλον πρέπει να ελευθερώνει τη δυναμικά δεσμευμένη μνήμη του, να σκοτώνει όλες τις διεργασίες-παιδιά που είναι ακόμη ενεργές και να μην αφήνει ζόμπι (να περιμένει τον τερματισμό τους).

Παράλληλα με την υποστήριξη των εντολών χρήστη, το περιβάλλον πρέπει να λειτουργεί και ως ένας απλός χρονοδρομολογητής έτσι ώστε κάθε στιγμή να εκτελείται μόνο μία διεργασία-παιδί (αν υπάρχει), και κάθε διεργασία να εκτελείται για συγκεκριμένο μερίδιο χρόνου. Η επιλογή της επόμενης διεργασίας είναι κυκλική (round-robin). Η εναλλαγή διεργασιών πρέπει να γίνεται στέλνοντας σήματα SIGSTOP και SIGCONT στις αντίστοιχες διεργασίες, ενώ η εκπνοή του χρόνου της τρέχουσας διεργασίας πρέπει να ελέγχεται μέσω ενός «ξυπνητηριού» σε συνδυασμό με το σήμα SIGALRM. Επίσης, εναλλαγή πρέπει να γίνεται και όταν τερματίζεται η διεργασία που εκτελείται, ακόμα και αν δεν έχει εκπνεύσει ο χρόνος της.

**Απαιτήσεις υλοποίησης:** Πρέπει να ελέγχετε τις τιμές που επιστρέφουν όλες οι κλήσεις συστήματος και να δράτε ανάλογα. Επιτρέπεται να χρησιμοποιήσετε καθολικές μεταβλητές τύπου `volatile sig_atomic_t` εφόσον αυτές προσπελάζονται μέσα από χειριστές σημάτων. Επίσης επιτρέπεται να χρησιμοποιήσετε καθολική μεταβλητή για την κεφαλή της λίστας (ή πίνακα) όπου αποθηκεύετε τις πληροφορίες για τις διεργασίες. Άλλες καθολικές μεταβλητές, η χρήση `goto/gets` και η άμεση ή έμμεση χρήση μη επιτρεπτών κλήσεων συστήματος σε χειριστές σημάτων, απαγορεύονται. Γράψτε ευανάγνωστο, καλά τεκμηριωμένο κώδικα. Κώδικας που δεν ικανοποιεί τις παραπάνω απαιτήσεις θα απορριφθεί, χωρίς εξέταση.

### Σημαντικές ημερομηνίες:

Φροντιστήριο: **Πέμπτη 10/5/2017**, στην ώρα του μαθήματος

Προθεσμία παράδοσης: **Κυριακή 27/5/2017, 23:59**

Οδηγίες παράδοσης: στη σελίδα του μαθήματος.

## Συνιστώμενα στάδια ανάπτυξης

### Στάδιο 1:

Υλοποιήστε ένα πρόγραμμα το οποίο θα μπορεί να εκτελείται μέσα από το περιβάλλον σας:

- Το πρόγραμμα παίρνει ως ορίσματα `-m M -b B` όπου `M` είναι ο μέγιστος αριθμός επαναλήψεων που θα πραγματοποιεί το πρόγραμμα, και `B` μια τιμή 1 ή 0 που καθορίζει αν θα μπλοκάρεται το SIGUSR1 ή όχι.
- Το πρόγραμμα αρχικοποιεί σε 0 ένα μετρητή τον οποίο αυξάνει κατά ένα κάθε 5 δευτερόλεπτα, εκτυπώνοντας κάθε φορά το `pid` του, την τιμή του μετρητή και τη μέγιστη τιμή που δόθηκε ως όρισμα. Όταν ο μετρητής φτάσει στη μέγιστη τιμή, το πρόγραμμα τερματίζει.
- Αν στα ορίσματα έχει δοθεί επιλογή `-b 1`, τότε το πρόγραμμα πρέπει να μπλοκάρει το SIGUSR1 για τις πρώτες μισές επαναλήψεις και μετά να το ξεμπλοκάρει (για πάντα), *δ ι α φ ο ρ ε τ ι κ ά τ ο σ ή μ α SIGUSR1 θ α π ρ έ π ε ι ν α π α ρ α μ έ ν ε ι σ υ ν ε χ ώ ς μ η -μ π λ ο κ α ρ ι σ μ έ ν ο .*
- Αν το πρόγραμμα λάβει το SIGUSR1 σε χρονική στιγμή που αυτό δεν είναι μπλοκαρισμένο, τότε επαναφέρει τον μετρητή στην τιμή 0.

Αφού ολοκληρώσετε το πρόγραμμα, εκτελέστε το με διάφορα ορίσματα για να βεβαιωθείτε ότι λειτουργεί σωστά σε όλες τις περιπτώσεις. Μπορείτε να του στείλετε το σήμα SIGUSR1 από την κονσόλα μέσω της εντολής `kill (man 1 kill)`

### Στάδιο 2:

Υλοποιήστε τη δομή για την αποθήκευση της πληροφορίας για κάθε διεργασία που δημιουργεί το περιβάλλον σας. Θα χρειαστεί να αποθηκεύσετε το `pid`, το όνομα προγράμματος και τα ορίσματα του. Επίσης, θα πρέπει με κάποιο τρόπο να καταγράφετε ποια διεργασία τρέχει κάθε στιγμή.

Η δομή θα πρέπει να διευκολύνει την «κυκλική» εναλλαγή διεργασιών, π.χ., κυκλική λίστα ή πίνακας με την τεχνική της κυκλικής αποθήκης (ψάξτε στο web για “circular buffer”). Γράψτε συναρτήσεις για εισαγωγή νέου στοιχείου (πάντα στην «ουρά» της δομής), διαγραφή στοιχείου, εύρεση της διεργασίας που τρέχει, εύρεση της επόμενης διεργασίας, ανίχνευση άδειας δομής, εκτύπωση των περιεχομένων της λίστας, και ότι άλλο πιστεύετε ότι θα σας χρειαστεί.

Η δομή πρέπει να είναι δυναμικά επεκτάσιμη (χρειάζεται ιδιαίτερη προσοχή αν αποφασίσετε να χρησιμοποιήσετε έναν πίνακα με την τεχνική της κυκλικής αποθήκης). Σε κάθε περίπτωση, χρειάζεται προσοχή στο πως θα υλοποιηθούν οι παραπάνω λειτουργίες, καθώς και σε ποια σημεία του προγράμματος θα καλούνται, ώστε να μην υπάρχουν bugs λόγω της ταυτόχρονης εκτέλεσης των διεργασιών και της ασύγχρονης φύσης των σημάτων που χειρίζεται το περιβάλλον εκτέλεσης.

### Στάδιο 3:

Υλοποιήστε τις λειτουργίες που περιγράφονται στο πινακάκι της πρώτης σελίδας. Γράψτε ένα loop το οποίο σε κάθε επανάληψη περιμένει από τον χρήστη να γράψει μια «εντολή» στο πληκτρολόγιο.

Για την εκτύπωση των περιεχομένων μέσω της εντολής `list`, προτείνουμε κάτι σαν το παρακάτω:

```
pid: 2879, name: (./test , -m, 10, -b, 1)
pid: 2880, name: (./test , -m, 4, -b, 0) (R)
pid: 2881, name: (./firefox)
```

έτσι ώστε να φαίνεται το αναγνωριστικό της διεργασίας, το όνομα του προγράμματος που αυτή εκτελεί, η λίστα ορισμάτων του προγράμματος που έδωσε ο χρήστης, και η ένδειξη (R) για την «τρέχουσα» διεργασία.

Σε αυτό το στάδιο μην ασχοληθείτε με τη χρονοδρομολόγηση. Βεβαιωθείτε ότι προσθαφαιρούνται σωστά οι διεργασίες στη δομή σας, και ότι ανιχνεύετε την περίπτωση να τερματίσει κάποια διεργασία χωρίς να έχει δοθεί τέτοια εντολή μέσα από το περιβάλλον σας (π.χ., μπορείτε να στείλετε σήμα τερματισμού μέσω άλλου τερματικού).

Ελέγξτε την λειτουργία του περιβάλλοντος σας, συμπεριλαμβανομένης της αποστολής του σήματος SIGUSR1, πρώτα για μία και μετά για περισσότερες διεργασίες. Ακολουθώσ επιβεβαιώστε ότι το περιβάλλον σας χειρίζεται σωστά τη λήψη/προώθηση του σήματος SIGUSR1.

#### **Στάδιο 4:**

Υλοποιήστε ένα ξυπνητήρι το οποίο χτυπά κάθε 20 δευτερόλεπτα. Αφού βεβαιωθείτε ότι λειτουργεί σωστά (π.χ., εκτυπώνοντας ένα σύντομο μήνυμα) υλοποιήστε τη χρονοδρομολόγηση: Κάθε φορά που χτυπά το ξυπνητήρι, θα πρέπει να σταματάτε με SIGSTOP τη διεργασία που τρέχει (αν υπάρχει) και να συνεχίζετε με SIGCONT την επόμενη (αν υπάρχει - διαφορετικά συνεχίζει την εκτέλεση της η τρέχουσα διεργασία). Υποθέσετε ότι κανείς άλλος δε στέλνει τα σήματα SIGSTOP, SIGCONT στις διεργασίες.

**ΠΡΟΣΟΧΗ:** Αν το ξυπνητήρι χτυπήσει την ώρα που το περιβάλλον εκτέλεσης προγραμμάτων περιμένει για είσοδο από το πληκτρολόγιο, θα έχετε ανεπιθύμητα αποτελέσματα. Χρησιμοποιήστε το flag επανεκκίνησης κλήσεων συστήματος κατά την εγκατάσταση του χειριστή σήματος. Επίσης, ίσως χρειαστεί να μπλοκάρτε το SIGALRM σε κάποιες περιπτώσεις.

**ΠΡΟΣΟΧΗ:** Διαβάστε προσεκτικά το man page του signal στο section 7 (man 7 signal) και ειδικά τη λίστα συναρτήσεων που επιτρέπεται να χρησιμοποιούνται σε χειριστές σημάτων. Βεβαιωθείτε ότι δε χρησιμοποιείτε άλλες συναρτήσεις στους χειριστές σας, διαφορετικά μπορεί να προκύψουν απροσδιόριστες/απρόβλεπτες παρενέργειες.