



ECE 316 - Operating Systems and Networking Laboratory

Practical Assignment 5 (Due: 19/02/2020)

Deliverables: Your report and code should be sent via email to lab teaching assistants (aanast01@ucy.ac.cy or ahadji29@ucy.ac.cy) prior the assignment examination and must include the usual cover page. In your report, include only the pseudocode, not the actual code, with any comments and description you may need to add, as well as a typical scenario that you use to test your programs. Email subject line should only consist of "ECE316_2020_5". Naming format for the zip file: `lastName.zip` (or `lastName_lastName.zip` if a group). **Caution:** Remove the executables (.exe) from the files before you send them!

Παρατηρήσεις:

- Για την πέμπτη εργαστηριακή άσκηση θα χρειαστείτε να εγκαταστήσετε στο προσωπική σας υπολογιστή λειτουργικό σύστημα UNIX ή LINUX. Σε περίπτωση που δεν είναι δυνατή η εγκατάσταση του η άσκηση μπορεί να υλοποιηθεί στους Η/Υ του εργαστηρίου ΘΕΕ01-B141. Σε αυτή την περίπτωση ενημερώστε τους βοηθούς του μαθήματος έτσι ώστε να διευθετηθεί ο τρόπος εξετάσεις. Ένα πιθανό λειτουργικό σύστημα που μπορείτε να εγκαταστήσετε είναι το UBUNTU που μπορείτε να το κατεβάσετε από: <https://www.ubuntu.com/desktop/developers> και να το εγκαταστήσετε παράλληλα με το λειτουργικό Windows ακολουθώντας τις έξι εντολές <https://vitux.com/how-to-install-ubuntu-18-04-along-with-windows-10/>.
- Οι εντολές που θα χρησιμοποιηθούν είναι μοναδικές για λειτουργικά UNIX. Έτσι για την επιτυχή εκπλήρωση των ασκήσεων θα μπορούσατε να χρησιμοποιήσετε το λειτουργικό "Ubuntu" ή να ακολουθήσετε τα βήματα για την ενεργοποίηση του εργαλείου bash που προσφέρει παρόμοιες δυνατότητες με Unix/Linux λειτουργικά συστήματα σε περιβάλλον Windows 10. Για την ενεργοποίηση του προγράμματος χρησιμοποιήστε τον οδηγό εγκατάστασης που επισυνάπτεται. <https://ubuntu.com/tutorials/tutorial-ubuntu-on-windows#1-overview>

1. [35%] Στην άσκηση αυτή καλείστε να δημιουργήσετε τρεις ξεχωριστές διεργασίες χρησιμοποιώντας την εντολή «fork()». Όλες οι διεργασίες πρέπει να δημιουργούνται από την κύρια/πατρική διεργασία. Η κάθε διεργασία πρέπει να εκτυπώνει στην οθόνη 51 τουλάχιστον μηνύματα τα οποία περιλαμβάνουν τον αριθμό της και τον αριθμό του μηνύματος (e.g Process 1 message 1).

Ερωτήσεις:

- a. Τι παρατηρείτε στην σειρά της εκτύπωσης των μηνυμάτων?
- b. Σε τι αντιστοιχεί η τιμή που επιστρέφει η επιτυχής ολοκλήρωση της εντολής «fork()»?
- c. Μετά από τρεις συνεχόμενες κλήσεις της «fork()» σε ένα πρόγραμμα, πόσες συνολικά διεργασίες υπάρχουν?
- d. Η αλλαγή στις τιμές των μεταβλητών μιας διεργασίας επηρεάζει τις τιμές των μεταβλητών μιας άλλης διεργασίας του ίδιου προγράμματος?



2. [15%] Στην άσκηση αυτή επαναλάβετε την Άσκηση1 αλλά αυτήν την φορά χρησιμοποιήστε την εντολή “sleep()” για να εισάγετε καθυστέρηση στην εκτύπωση των μηνυμάτων. Η καθυστέρηση πρέπει να είναι μεταβλητή και διαφορετική για κάθε διεργασία. Η καθυστέρηση κάθε διεργασίας θα εξαρτάται από τον αριθμό της διεργασίας και τον αριθμό του μηνύματος.

Για παράδειγμα: Η διεργασία 2 για το πέμπτο μήνυμα θα καθυστερεί $2 * 5 * 100\text{ms}$.

3. [40%] Δημιουργήστε ένα πρόγραμμα με 3 διεργασίες που η κάθε μία εκτυπώνει συνολικά 51 μηνύματα. Χρησιμοποιώντας τις σωληνώσεις (μία σωλήνωση για κάθε διεργασία) καθορίστε την σειρά εκτύπωσης των διεργασιών οι οποίες τυπώνουν μόνο ένα μήνυμα σε κάθε κύκλο. Για παράδειγμα αν επιλέξουμε να εκτυπώνει ένα μήνυμα η διεργασία 2, ένα η διεργασία 1 και ένα η 3 σε κάθε κύκλο, τότε το αποτέλεσμα θα είναι το ακόλουθο:

Process 2 message 1
Process 1 message 1
Process 3 message 1
Process 2 message 2
Process 1 message 2
Process 3 message 2

...

Βοήθεια:

Η πατρική διεργασία θα πρέπει να γράφει στην σωλήνωση από την οποία διαβάζει η διεργασία 2. Η διεργασία 2 θα διαβάζει την σωλήνωση αυτήν και θα γράφει στην επόμενη κ.ο.κ. Επίσης σε κάθε ανάγνωση και γραφή όλες οι σωληνώσεις που δεν χρησιμοποιούνται πρέπει να είναι κλειστές.

4. [10%] Στην άσκηση αυτή καλείστε να αλλάξετε τον κώδικα της άσκησης 3 έτσι ώστε να μπορεί να δεχτεί N διεργασίες. Ο ακριβής αριθμός των διεργασιών, N, να καθορίζεται από το πρόγραμμα μέσω μιας σταθερής.

Ερωτήσεις:

- Πως καταλαβαίνει μία διεργασία το πότε πρέπει να διαβάσει από μία σωλήνωση?
- Πιο πρόβλημα προκύπτει όταν η πατρική διεργασία τελειώσει πριν τις υπόλοιπες διεργασίες του προγράμματος?