ECE 316 - Operating Systems and Networking Laboratory

Practical Assignment 5

General Instructions

For each Assignment, a report (.pdf) and the source-code (e.g., .c, .cpp, .m, .bat etc.) of the solution should be submitted through Microsoft Teams "ECE316 – Operating Systems and Networks Laboratory" no later than the due date of the Assignment. The report should start with a cover page that clearly contains the assignment number and the Team number and the names of the members. In your report, include only the pseudocode, not the actual code, with any comments and description you may want to add, as well as a typical scenario that you used to test your programs. Please note that the report should be as concise as possible. Caution: You are not allowed to upload executables (.exe)!

If input test files are given, you are not allowed to make any changes to the provided input files.

Report File Naming Format: "Team#_Assignment#.pdf"

Βοηθητικές Οδηγίες:

- Για την πέμπτη άσκηση θα χρειαστείτε να εγκαταστήσετε στο προσωπικό σας υπολογιστή λειτουργικό σύστημα UNIX ή LINUX. Ένα πιθανό λειτουργικό σύστημα που μπορείτε να εγκαταστήσετε είναι το UBUNTU που μπορείτε να το κατεβάσετε από:
  - (Προτεινόμενο)Επιλογή 1: Εγκατάσταση του Ubuntu μέσω WSL2 σε Windows 10+ https://ubuntu.com/wsl.
  - Επιλογή 2: https://www.ubuntu.com/desktop/developers και να το εγκαταστήσετε παράλληλα με το λειτουργικό Windows ακολουθώντας τις εξής εντολές https://vitux.com/how-to-install-ubuntu-18-04-along-with-windows-10/.
- Τα scripts που γράφετε πρέπει αναγνωρίζουν τυχόν λανθασμένες εισόδους και να εμφανίζουν τα κατάλληλα μηνύματα λάθους.
- Στα πιο κάτω links βρίσκεται βοηθητικό υλικό για τη διεκπεραίωση της άσκησης:
  - https://en.wikibooks.org/wiki/Bash_Shell_Scripting
  - http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_02_01.html
  - http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html

Description:

1. **Άσκηση 1 [35%]**
   Στην άσκηση αυτή καλείστε να δημιουργήσετε τέσσερεις 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 τουλάχιστον 89 μηνύματα τα οποία περιλαμβάνουν τον αριθμό της και τον αριθμό του μηνύματος (e.g., Process 1 message 1).

Η κάθε διεργασία θα πρέπει επίσης να φυλάει και να εκτυπώνει στο τέλος του προγράμματος, τους χρόνους εκτέλεσης σε δευτερόλεπτα για 1) την εκτύπωση του πρώτου μηνύματος 2) την εκτύπωση του τελευταίου μηνύματος και 3) το συνολικό χρόνο που χρειάστηκε για να εκτυπώσει όλα τα μηνύματα (e.g., Process 1: time of first message/ time of last message/ duration of printing).

Ερωτήσεις:

a. Τι παρατηρείτε στην σειρά εκτύπωσης των μηνυμάτων;
b. Σε τι αντιστοιχεί η τιμή που επιστρέφει η επιτυχής ολοκλήρωση της εντολής «fork()»;
c. Μετά από τέσσερις 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; Εξηγείστε την απάντηση σας.
e. Τι παρατηρείτε στους χρόνους εκτέλεσης μεταξύ των διεργασιών και μεταξύ διαφορετικών εκτελέσεων του ίδιου προγράμματος;

2. Άσκηση 2 [15%]

Στην άσκηση αυτή επαναλάβετε την Άσκηση 1, αλλά αυτήν την φορά χρησιμοποιήστε την εντολή "sleep()" για να εισάγετε καθυστέρηση στην εκτύπωση των μηνυμάτων. Η καθυστέρηση πρέπει να είναι μεταβαλλόμενη 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∗0.05 seconds.

3. Άσκηση 3 [40%]

Δημιουργήστε ένα πρόγραμμα με 4 διεργασίες που η κάθε μία εκτυπώνει συνολικά 100 μηνύματα. Χρησιμοποιώντας τις σωληνώσεις/pipes (μία σωλήνωση για κάθε διεργασία), καθορίστε την σειρά εκτύπωσης των διεργασιών, έτσι ώστε να τυπώνουν μόνο τρία μηνύματα σε κάθε κύκλο. Για παράδειγμα, αν επιλέξουμε να εκτυπώνει τρία μηνύματα η διεργασία 3, τρία η διεργασία 2, τρία η διεργασία 1 και τρία η διεργασία 4 σε κάθε κύκλο, τότε το αποτέλεσμα θα είναι το ακόλουθο:

Process 3 message 1
Process 3 message 2
Process 3 message 3
Process 2 message 1
Process 2 message 2
Process 2 message 3
Process 1 message 1
Process 1 message 2
Process 1 message 3
Process 4 message 1
Process 4 message 2
Process 4 message 3
Process 3 message 4
Process 3 message 5
Process 3 message 6
Process 2 message 4
Process 2 message 5
Process 2 message 6
Process 1 message 4
Process 1 message 5
Process 1 message 6
Process 4 message 4
Process 4 message 5
Process 4 message 6
…

Βοήθεια: Η πατρική διεργασία θα πρέπει να γράφει στην σωλήνωση από την οποία διαβάζει η διεργασία 3. Η διεργασία 3 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. **Άσκηση 4 [10%]**

Στην άσκηση αυτή καλείστε να αλλάξετε τον κώδικα της άσκησης 3 έτσι ώστε να μπορεί να δεχτεί Ν διεργασίες που θα τυπώνουν Μ μηνύματα σε κάθε κύκλο. Οι ακριβείς αριθμοί των Ν διεργασιών και Μ μηνυμάτων, θα καθορίζονται από το πρόγραμμα μέσω δύο σταθερών τιμών.

Ερωτήσεις:

a. Πως καταλαβαίνει μία διεργασία το πότε πρέπει να διαβάσει από μία σωλήνωση;
b. Ποιό πρόβλημα προκύπτει όταν η πατρική διεργασία τελειώσει πριν τις υπόλοιπες διεργασίες του προγράμματος;
c. Πόσος χρόνος σε δευτερόλεπτα περνά από την στιγμή που τελειώνει η πρώτη διεργασία μέχρι να τελειώσει και η τελευταία διεργασία;