

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Υποχρεωτικό Μάθημα 4^{ου} εξαμήνου

Εαρινό Εξάμηνο 2016-2017

1^η Προγραμματιστική Εργασία

Αντικείμενο: Στα πλαίσια της 1^{ης} εργασίας καλείστε να υλοποιήσετε ένα δικό σας φλοιό (`mysh`) που θα μιμείται το φλοιό του Linux, δηλαδή, ένα πρόγραμμα το οποίο θα επαναλαμβάνει τα εξής βήματα: α) εμφάνιση μιας προτροπής (`prompt`), β) ανάγνωση μιας γραμμής εντολών (`command line`) από το τερματικό, και γ) δημιουργία των κατάλληλων διεργασιών προς εκτέλεση της γραμμής εντολών. Η εργασία είναι οργανωμένη σε στάδια κλιμακωτής δυσκολίας που θα βαθμολογηθούν ανεξάρτητα, έτσι ώστε ακόμη κι αν δεν ολοκληρώσετε όλα τα ζητούμενα, να πάρετε βαθμούς. Στο x-οστό στάδιο θα δημιουργήσετε έναν φλοιό με όνομα `mysh<x>`, επεκτείνοντας τη λειτουργικότητα του φλοιού του προηγούμενου βήματος `mysh<x-1>`. Ο κώδικάς σας θα πρέπει να λειτουργεί σωστά στην εικονική μηχανή που διατίθεται στα CSLAB και στο διαδίκτυο (βλ. ανακοίνωση στο `eclass`). Η εργασία είναι ομαδική και είναι σχεδιασμένη για ομάδες τριών (3) ατόμων, μπορείτε όμως να την κάνετε και σε μικρότερες ομάδες.

Βήματα εργασίας:

1. `mysh1` (20%). Ο φλοιός αυτός θα δέχεται απλές μεμονωμένες εντολές. Συγκεκριμένα, α) θα εμφανίζει ως προτροπή το όνομά του και τον χαρακτήρα «>» (δηλαδή, `mysh1>`), β) θα διαβάζει το όνομα ενός προγράμματος από το τερματικό, γ) θα δημιουργεί μια νέα διεργασία για να εκτελέσει το πρόγραμμα το οποίο διάβασε, και δ) θα περιμένει να τερματιστεί η νέα διεργασία. Μόλις η νέα διεργασία τερματιστεί, ο φλοιός θα διαβάζει και θα εκτελεί την επόμενη εντολή. Για παράδειγμα, ο φλοιός θα μπορεί να εκτελέσει γραμμές εντολών της μορφής «`ls`».
2. `mysh2` (20%). Ο φλοιός αυτός θα κάνει ό,τι και ο `mysh1` και επιπλέον θα δέχεται παραμέτρους για τις εντολές που εκτελεί. Για παράδειγμα, ο φλοιός θα μπορεί να εκτελέσει γραμμές εντολών της μορφής «`ls -l`».
3. `mysh3` (20%). Ο φλοιός αυτός θα κάνει ό,τι και ο `mysh2` και επιπλέον θα δέχεται ανακατεύθυνση της τυπικής εισόδου/τυπικής εξόδου από/προς αρχεία για τις εντολές που εκτελεί. Για παράδειγμα, ο φλοιός θα μπορεί να εκτελέσει γραμμές εντολών της μορφής «`sort -u < in.txt > out.txt`».
4. `mysh4` (20%). Ο φλοιός αυτός θα κάνει ό,τι και ο `mysh3` και επιπλέον θα εκτελεί ακολουθίες εντολών με μία σωλήνωση. Για παράδειγμα, ο φλοιός θα μπορεί να εκτελέσει γραμμές εντολών της μορφής «`ls -l /home/csuser/Downloads | sort -u > listing.txt`».
5. `mysh5` (20%). Ο φλοιός αυτός θα κάνει ό,τι και ο `mysh4` και επιπλέον θα εκτελεί ακολουθίες εντολών χωρίς περιορισμό στο πλήθος των σωληνώσεων. Για παράδειγμα, ο φλοιός θα μπορεί να εκτελέσει γραμμές εντολών της μορφής «`ls -l /home/csuser/Downloads | sort -u | wc -l > count.txt`».

Πρόσθετες απαιτήσεις:

- Απαγορεύεται η χρήση κλήσεων συστήματος που δημιουργούν νέο φλοιό για εκτέλεση των εντολών (π.χ. εκτέλεση του `/bin/sh`), καθώς και της κλήσης `system()`. Χρήση των προαναφερθέντων επιφέρει μηδενισμό.
- Θα πρέπει να είναι σαφές ποιος φλοιός εκτελείται. Για παράδειγμα, αν τρέχει ο πρώτος φλοιός, στην αρχή κάθε γραμμής πρέπει να εμφανίζεται «`mysh1>`».

- Για την ανακατεύθυνση εξόδου θα πρέπει να υποστηρίζεται τόσο η δημιουργία καινούργιου αρχείου (>), όσο και η προσάρτηση (>>) της εξόδου σε ήδη υπάρχων αρχείο. Για παράδειγμα αν το αρχείο out.txt προϋπήρχε και εκτελούσαμε την εντολή `sort -u < in.txt > out.txt` τα περιεχόμενα του θα διαγραφούν, αντιθέτως αν εκτελούσαμε την εντολή `sort -u < in.txt >> out.txt` το αποτέλεσμα θα προστεθεί στο τέλος του αρχείου out.txt.
- Ο φλοιός σας πρέπει να τερματίζει όταν διαβάσει τον ειδικό χαρακτήρα EOF. Μπορείτε να στείλετε EOF στον φλοιό πληκτρολογώντας CTRL-D στην αρχή μιας εντολής.
- Θα πρέπει να ορίσετε ένα μέγιστο συνολικό μέγεθος για τη γραμμή εντολών (τουλάχιστον 255 χαρακτήρες) και να ελέγχετε ότι η είσοδος δεν το ξεπερνά.
- Ο φλοιός σας δεν πρέπει να αφήνει διεργασίες ζόμπι (θα βαθμολογούνται αρνητικά).
- Ο κώδικάς σας πρέπει να είναι δομημένος με συναρτήσεις με συγκεκριμένο έργο, έτσι ώστε να μπορείτε να τον επαναχρησιμοποιήσετε σε πολλούς φλοιούς.
- Οι συναρτήσεις που χρησιμοποιούνται σε πολλούς φλοιούς πρέπει να βρίσκονται σε ένα βοηθητικό αρχείο κώδικα, και όχι να αντιγράφονται στο αρχείο κώδικα κάθε φλοιού.
- Θα πρέπει να γίνονται όλοι οι κατάλληλοι έλεγχοι σε συναρτήσεις κλήσης συστήματος (π.χ. `fork()`, `pipe()`) καθώς και συναρτήσεις δέσμευσης μνήμης.

Υποδείξεις:

- Για να αναλύσετε τη γραμμή εντολών σε εντολές και παραμέτρους, μπορείτε να χρησιμοποιήσετε τη συνάρτηση `char * strtok (char *restrict newstring, const char *restrict delimiters)`. Προσοχή στο ότι η `strtok()` τροποποιεί την πρώτη της παράμετρο.
- Για να συνδεθεί αυτόματα το βοηθητικό αρχείο κώδικα με το κύριο αρχείο κώδικα κάθε φλοιού, αρκεί να τα γράψετε και τα δύο στη γραμμή εντολών του μεταγλωττιστή.
- Για να συνδέσετε τις εντολές με τις σωληνώσεις και για να κάνετε ανακατεύθυνση εισόδου/εξόδου, θα χρειαστείτε την κλήση `dup ()` ή την κλήση `dup2 ()`.
- Για να εντοπίσετε το τέλος της εισόδου, βρείτε τι επιστρέφεται στο πρόγραμμα όταν ο χρήστης στέλνει EOF.
- Δεν θα υλοποιήσετε τις ενσωματωμένες εντολές στο φλοιό (π.χ. δομές ελέγχου και επανάληψης) που αποτελούν μέρος του κώδικα του φλοιού. Ο φλοιός σας αρκεί να εκτελεί μόνο εξωτερικές εντολές, δηλαδή προγράμματα που καλούνται από τον φλοιό.
- Σκεφτείτε κατά το σχεδιασμό του κάθε φλοιού:
 - Αν και πόσες διεργασίες-παιδιά χρειάζονται.
 - Αν πρέπει και πώς μπορείτε να επιβάλλετε τη σειρά εκτέλεσης των παιδιών.
 - Πόσες σωληνώσεις χρειάζονται.
 - Αν και πώς μπορεί να αξιοποιηθεί ο κώδικας από προηγούμενα βήματα.

Παραδοτέα: Ο κώδικάς σας πρέπει να αποτελείται από ένα αρχείο με δηλώσεις για όλους τους φλοιούς, ένα αρχείο κώδικα C με τις κοινές συναρτήσεις για όλους τους φλοιούς και από ένα αρχείο κώδικα C για κάθε φλοιό. Τα αρχεία αυτά πρέπει να έχουν ονόματα της μορφής `p3x-p3y-p3z-mysh.h` για τις δηλώσεις, `p3x-p3y-p3z-mysh-common.c` για τον κώδικα C των κοινών συναρτήσεων και `p3x-p3y-p3z-myshN.c` για τον κώδικα C κάθε φλοιού, όπου N είναι ο αριθμός του φλοιού από 1 έως 5 και `p3x-p3y-p3z` είναι οι αριθμοί μητρώου σας. Εκτός από τον κώδικα, θα πρέπει να γράψετε μία αναφορά η οποία να περιγράφει τη δομή του κώδικά σας για κάθε φλοιό και να αναφέρει τυχόν περιορισμούς των φλοιών σας ή τυχόν πρόσθετα χαρακτηριστικά που έχετε υλοποιήσει. Η αναφορά πρέπει να είναι ένα αρχείο σε μορφή PDF με όνομα της μορφής `p3x-p3y-p3z-mysh.pdf`. Αυτά τα οκτώ αρχεία (και τίποτα άλλο) θα πρέπει να συμπιεστούν σε ένα αρχείο σε μορφή 7zip με όνομα της μορφής `p3x-p3y-p3z-mysh.7z` και να υποβληθούν από ένα μόνο μέλος της ομάδας μέσω της υποβολής εργασιών του eclass.

Προθεσμία υποβολής: Τα συμπιεσμένα αρχεία με τις εργασίες σας θα πρέπει να παραδοθούν μέσω του eclass μέχρι την Δευτέρα 8/5/2017 και ώρα 23:59.

Βαθμολόγηση και εξέταση: Αρχεία που δεν θα έχουν την *ακριβή* ονοματολογία που αναφέρεται παραπάνω *θα μηδενιστούν*, διότι η μεταγλώττιση και ο έλεγχος λειτουργικότητας θα γίνουν με αυτοματοποιημένο τρόπο που βασίζεται στη συγκεκριμένη ονοματολογία. Εργασίες με ομοιότητες που υποδεικνύουν αντιγραφή *θα μηδενιστούν όλες* (θα γίνει έλεγχος με ειδικό πρόγραμμα). Δειγματοληπτικά, θα κληθούν ορισμένες ομάδες για επίδειξη των εργασιών και προφορική εξέταση στα CSLAB. Μέλη ομάδων, ή και ολόκληρες ομάδες, που δεν θα παρευρεθούν στην προφορική εξέταση, *θα μηδενιστούν*.