

**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ**  
**Εξετάσεις Β' Περιόδου 2007 (26/9/2007)**

1. (α') Τι δικαιώματα (permissions) απαιτούνται για να μπορεί να διαγραφεί ένα αρχείο, δίνοντας γι' αυτό απόλυτο ή σχετικό όνομα-μονοπάτι; Τι δικαιώματα απαιτούνται για να μπορεί να εκτελεσθεί επιτυχώς η εντολή "ls dir", όταν ο κατάλογος dir βρίσκεται στον τρέχοντα κατάλογό μας; Χρειάζονται περισσότερα δικαιώματα ή όχι και, αν ναι, ποια, αν η εντολή είναι η "ls -l dir";  
(β') Έστω ότι θέλετε να υλοποιήσετε μία εφαρμογή στην οποία μία διεργασία  $P1$  πρέπει να στέλνει δεδομένα σε μία διεργασία  $P2$  και μία διεργασία  $P3$  να στέλνει δεδομένα σε μία διεργασία  $P4$ . Κάτω από ποιες προϋποθέσεις θα μπορούσε η προηγούμενη επικοινωνία να επιτευχθεί μέσω σωλήνων; Θα αρκούσε ένας σωλήνας ή χρειάζονται περισσότεροι; Θα μπορούσαν να χρησιμοποιηθούν υποδοχές ροής για να επιτευχθεί αυτή η επικοινωνία και, αν ναι, σε ποιο πεδίο; Θα μπορούσατε να χρησιμοποιήσετε ουρές μηνυμάτων; Αν ναι, αρκεί μία ή χρειάζονται περισσότερες; Πώς θα σχολιάζατε τη χρήση κοινής μνήμης για την επικοινωνία αυτή;
2. Γράψτε ένα πρόγραμμα για το κέλυφος Bourne (έστω ότι ονομάζεται "splitinp") το οποίο να διαβάζει από την προκαθορισμένη είσοδο (stdin) γραμμές κειμένου και τις οποίες να αποθηκεύει, με κυκλικό τρόπο, σ' ένα σύνολο από αρχεία με ονόματα της μορφής " $\langle prefix \rangle$ - $\langle partno \rangle$ ". Το πλήθος των αρχείων είναι  $\langle N \rangle$ , το οποίο είτε δίνεται σαν πρώτο όρισμα στο πρόγραμμα "splitinp", είτε, αν δεν δοθεί, ορίστε την προκαθορισμένη τιμή  $\langle N \rangle = 5$ . Το  $\langle prefix \rangle$  είτε δίνεται σαν δεύτερο όρισμα στο πρόγραμμα "splitinp" (φυσικά, αν έχει δοθεί και πρώτο όρισμα) είτε, αν δεν δοθεί, ορίστε την προκαθορισμένη τιμή  $\langle prefix \rangle = file$ . Με τον όρο "κυκλικός τρόπος" αποθήκευσης της εισόδου εννοείται ότι η πρώτη γραμμή της εισόδου θα αποθηκευθεί στο αρχείο " $\langle prefix \rangle$ -1", η δεύτερη στο " $\langle prefix \rangle$ -2", η  $\langle N \rangle$ -οστή στο " $\langle prefix \rangle$ - $\langle N \rangle$ ", η  $(\langle N \rangle + 1)$ -οστή στο " $\langle prefix \rangle$ -1" και ούτω καθεξής. Δηλαδή, το  $\langle partno \rangle$  μεταβάλλεται κυκλικά από την τιμή 1 έως την τιμή  $\langle N \rangle$  κατά την αποθήκευση των γραμμών εισόδου. Κάποια παραδείγματα εκτέλεσης είναι τα εξής:

```
$ cat tosplit
This is the first line
And this is the 2nd one
    This is the third line
And line no 4

    Line no 6 (previous was empty)
* * And the last, the seventh * *
$ ./splitinp < tosplit
$ cat file-1
This is the first line
    Line no 6 (previous was empty)
$ cat file-2
And this is the 2nd one
* * And the last, the seventh * *
$ cat file-3
    This is the third line
$ cat file-4
And line no 4
$ cat file-5

$ /bin/rm file-*
$ ./splitinp 3 < tosplit
```

```
$ cat file-1
This is the first line
And line no 4
* * And the last, the seventh * *
$ cat file-2
And this is the 2nd one

$ cat file-3
    This is the third line
    Line no 6 (previous was empty)
$ /bin/rm file-*
$ ./splitinp 2 splitted < tosplit
$ cat splitted-1
This is the first line
    This is the third line

* * And the last, the seventh * *
$ cat splitted-2
And this is the 2nd one
And line no 4
    Line no 6 (previous was empty)
$ /bin/rm splitted-*
$
```

3. (α') "Όταν εκτελούμε ένα πρόγραμμα κάτω από ένα κέλυφος (στο προσκήνιο), το κέλυφος αυτό δημιουργεί ...**(A)**..., μετά ...**(B)**... και, τέλος, ...**(Γ)**... πριν επιστρέψει το `prompt` για αναμονή νέας εντολής." Συμπληρώστε κατάλληλα τα τμήματα **(A)**, **(B)** και **(Γ)** της πρότασης αυτής. Διατυπώστε την αντίστοιχη πρόταση όταν εκτελούμε κάτω από ένα κέλυφος μία σωλήνωση μεταξύ δύο ή περισσότερων προγραμμάτων.

(β') Παρατηρήστε την παρακάτω διαλογική επικοινωνία και συμπληρώστε στα αποσιωπητικά πιθανά αποτελέσματα, δικαιολογώντας τα με βάση και τις απαντήσεις σας στο προηγούμενο ερώτημα. Εξηγήστε οπωσδήποτε πότε αποφασίζει το κέλυφος να εκτυπώσει το `prompt` για αναμονή νέας εντολής, μετά από την εκτέλεση της σωλήνωσης. Αν έλειπε από το παρακάτω πρόγραμμα `3fw.c` η δεύτερη `printf`, σε τι θα διέφεραν τα αποτελέσματα; Υπήρχε περίπτωση τότε να συμβεί κάτι περίεργο με την εκτύπωση του `prompt`;

```
$ cat 3fw.c
#include <stdio.h>
main()
{ int s;
  fork(); fork(); fork();
  printf("%d %d\n", getpid(), getppid());
  printf("%d %d %d\n", wait(&s), wait(&s), wait(&s)); }
$ cc -o 3fw 3fw.c
$ ./3fw
.....
$ ./3fw | wc -l
.....
$
```

4. Σε μία πολυνηματική εφαρμογή, το αρχικό νήμα τοποθετεί στοιχεία προς επεξεργασία σε μία καθολικά προσπελάσιμη ουρά και, στη συνέχεια, δημιουργεί `NWorkers` νήματα-εργάτες, τα οποία διαχειρίζονται την ουρά για να επεξεργασθούν τα δεδομένα που περιέχει. Κάθε εργάτης, όταν δεν έχει δουλειά, εξάγει κάποιο στοιχείο από την ουρά, το επεξεργάζεται και, αναλόγως με το αποτέλεσμα της επεξεργασίας, ενδέχεται να προσθέσει στην ουρά και κάποιο νέο στοιχείο για επεξεργασία. Μετά, συνεχίζει παίρνοντας νέο στοιχείο από την ουρά για επεξεργασία. Η κοινή προσπάθεια των εργατών τερματίζει όταν δεν υπάρχει, και είναι βέβαιο ότι δεν θα υπάρξει, άλλο στοιχείο στην ουρά για επεξεργασία. Το απόσπασμα κώδικα C που φαίνεται στη συνέχεια προέρχεται από τη συνάρτηση των νημάτων-εργατών. Συμπληρώστε το απόσπασμα αυτό, όπου υπάρχουν αποσιωπητικά, για να επιτευχθεί η επιθυμητή συμπεριφορά. Μπορείτε να χρησιμοποιήσετε και επιπλέον, εξωτερικές ή όχι, μεταβλητές, όχι όμως σηματοφόρους ή μεταβλητές συνθήκης, πλην αυτών που περιλαμβάνονται ήδη στο απόσπασμα. Ότι συναρτήσεις χρησιμοποιούνται, θεωρήστε ότι έχουν οριστεί κατάλληλα.

```
while (!pthread_mutex_lock(&mtx)) {
  if (is_empty_queue(queue)) {
    .....
    pthread_cond_wait(&cvar, &mtx);
    .....
    continue; }
  item = remove_from_queue(queue);
  .....
  newitem = process_item(item);
  if (needs_processing(newitem)) {
    .....
    insert_in_queue(queue, newitem);
    ..... } }
```