

**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ**  
**Εξετάσεις Α' Περιόδου 2008 (4/2/2008)**

ΟΝΟΜΑ: \_\_\_\_\_ ΕΠΩΝΥΜΟ: \_\_\_\_\_  
ΚΩΔΙΚΟΣ: \_\_\_\_\_ EMAIL: \_\_\_\_\_ ΕΞΑΜΗΝΟ: \_\_\_\_\_

**ΑΣΚΗΣΗ 1 (28%)**

Α. Γράψτε ένα πρόγραμμα `ineqJoin` σε κέλυφος `Bash` που να υλοποιεί μία συνένωση δύο αρχείων όπως περιγράφεται παρακάτω. Τα ονόματα των δύο αρχείων να δίνονται ως παράμετροι εισόδου του προγράμματος (δηλαδή το πρόγραμμα σας να καλείται ως `./ineqJoin file1 file2`). Τα δύο αρχεία περιέχουν διαφορετικό αριθμό γραμμών. Κάθε γραμμή σε κάθε αρχείο περιέχει 2 θετικούς ακέραιους αριθμούς που διαχωρίζονται από το χαρακτήρα '@'. Το ζητούμενο είναι το πρόγραμμα `ineqJoin` να τυπώνει στην προκαθορισμένη έξοδο όλα τα ζεύγη γραμμών από τα δύο αρχεία όπου ο πρώτος αριθμός της γραμμής του `file1` να είναι μικρότερος του πρώτου αριθμού της γραμμής του `file2`. Οι 4 ακέραιοι που αντιστοιχούν σε κάθε ζεύγος γραμμών να διαχωρίζονται από ένα κενό χαρακτήρα. Για παράδειγμα, μία γραμμή του `file1` `5@10` μπορεί να συνενωθεί με μία γραμμή `7@1` του `file2`, αφού `5 < 7`, οπότε τυπώνεται η γραμμή `"5 10 7 1"`. Μία επιπλέον απαίτηση είναι οι γραμμές που εκτυπώνονται στην προκαθορισμένη έξοδο να είναι ταξινομημένες ανά αύξοντα τιμή του πρώτου αριθμού που εμφανίζεται στη γραμμή. Δύο παραδείγματα εκτέλεσης του προγράμματος `ineqJoin` απεικονίζονται στο Σχήμα 1.

```
iokasti:/home/users/adeli>cat file1
11@1341
51@1
1231@1254
1@1
31@100
42@200
iokasti:/home/users/adeli>cat file2
22@231
42@245
2@22
iokasti:/home/users/adeli>./ineqJoin file1 file2
1 1 22 231
1 1 42 245
1 1 2 22
11 1341 22 231
11 1341 42 245
31 100 42 245
iokasti:/home/users/adeli>./ineqJoin file2 file1
2 22 11 1341
2 22 51 1
2 22 1231 1254
2 22 31 100
2 22 42 200
22 231 51 1
22 231 1231 1254
22 231 31 100
22 231 42 200
42 245 51 1
42 245 1231 1254
```

Σχήμα 1

Γραμμή	
1	<code>#include &lt;stdio.h&gt;</code>
2	<code>#include &lt;sys/types.h&gt;</code>
3	<code>#include &lt;fcntl.h&gt;</code>
4	<code>int main(int argc, char *argv[]) {</code>
5	<code>int pid, fd, pd[2];</code>
6	<code>char c;</code>
7	<code>pipe(pd);</code>
8	<code>dup2(pd[0], 0);</code>
9	<code>close(pd[0]);</code>
10	<code>if ( (pid = fork() ) == -1) {</code>
11	<code> perror("fork"); exit(-1); }</code>
12	<code>if (pid == 0) {</code>
13	<code> close(0);</code>
14	<code> fd = open(argv[1], O_RDONLY);</code>
15	<code> dup2(pd[1], 1);</code>
16	<code> close(pd[1]);</code>
17	<code> execlp("wc", "wc", NULL);</code>
18	<code> printf("After execlp\n");</code>
19	<code> }</code>
20	<code> else {</code>
21	<code> close(pd[1]);</code>
22	<code> while (read(0, &amp;c, 1) &gt; 0)</code>
23	<code> write(1, &amp;c, 1);</code>
24	<code> }</code>
25	<code> return 1;</code>
26	<code> }</code>

Σχήμα 2

B. Χωρίς να γράψετε κώδικα, εξηγήστε αν το πρόγραμμά σας θα μπορούσε να υλοποιήσει την παραπάνω συνένωση στην περίπτωση που η κάθε γραμμή του file2 περιείχε μεταβλητό αριθμό αριθμών/στηλών. Αν όχι, περιγράψτε σύντομα ποιες γραμμές του προγράμματος θα έπρεπε να μεταβληθούν. Θα έπρεπε να αλλάζετε δηλαδή όλο το πρόγραμμά σας, ή απλά να χρησιμοποιήσετε κάποιες άλλες εντολές (και ποιες) σε ένα κομμάτι του;

### ΑΣΚΗΣΗ 2 (30%)

A. Έστω ότι σχεδιάζετε μία εφαρμογή όπου θα μεταδίδετε ζωντανά ποδοσφαιρικούς αγώνες μέσω του διαδικτύου, κάνοντας χρήση επικοινωνίας βασισμένης σε υποδοχές. Τι είδους υποδοχές θα προτιμούσατε ως βάση για μία τέτοια εφαρμογή: υποδοχές ροής ή τηλεγραφικές υποδοχές; Δικαιολογήστε την απάντησή σας.

B. Έστω ότι ο χρήστης srgo εκτελεί ένα πρόγραμμα prog1, και ένας διαφορετικός χρήστης adeli εκτελεί ένα πρόγραμμα prog2. Με ποιες από τις παρακάτω μεθόδους μπορούν να ανταλλάξουν δεδομένα τα prog1 και prog2, και κάτω από ποιες προϋποθέσεις: (α) Σωλήνες; (β) Υποδοχές ροής; (γ) Τηλεγραφικές υποδοχές; (δ) Ουρές Μηνυμάτων; (ε) Κοινή Μνήμη; (στ) Σύνολα σηματοφόρων.

Γ. Μία διεργασία δημιουργεί μία ουρά μηνυμάτων, και στη συνέχεια δημιουργεί 10 διεργασίες παιδιά. Κάθε διεργασία παιδί θέλει να στείλει ένα ξεχωριστό μήνυμα με ένα διαφορετικό τυχαίο αριθμό (1) σε κάθε ένα από τα αδέρφια του; και (2) στον πατέρα του. Χωρίς να γράψετε την υλοποίηση όλου του προγράμματος, εξηγήστε αλγοριθμικά πώς θα υλοποιούσατε το παραπάνω πρόγραμμα. Ορίστε τις μεταβλητές που θα χρησιμοποιούσατε, εξηγήστε πώς θα βάζατε πληροφορία σε αυτές τις μεταβλητές και πώς θα επιτυγχάνατε την επικοινωνία μεταξύ όλων των διεργασιών χρησιμοποιώντας τη μόνη ουρά μηνυμάτων.

### ΑΣΚΗΣΗ 3 (27%)

Εκατομμύρια κόσμος κατεβάζουν αρχεία βίντεο και ήχου. Αποφασίζετε ότι θα χρησιμοποιήσετε τις γνώσεις του μαθήματος για να φτιάξετε ένα δικό σας C πρόγραμμα που θα κατεβάζει τραγούδια και βίντεο. Η ακόλουθη συνάρτηση έχει ήδη γραφεί για εσάς και είναι νηματικά ασφαλής. Κατεβάζει τα περιεχόμενα του καθορισμένου αρχείου από το δηλωμένο εξυπηρετητή και επιστρέφει το μέγεθος του αρχείου.

```
int DownloadFile(const char *server, const char *file);
```

Σκοπός σας είναι να υλοποιήσετε μία συνάρτηση **DownloadLibrary** (της οποίας τα ορίσματα θα τα καθορίσετε εσείς) η οποία να δημιουργεί ένα νήμα για κάθε αρχείο που θέλουμε να κατεβάσουμε και μετά (αφού έχει δημιουργήσει όλα τα νήματα) να περιμένει όλα τα νήματα να τερματίσουν. Κάθε νήμα κατεβάζει το αρχείο που του έχουμε αναθέσει, και προσθέτει το μέγεθός του σε μία μεταβλητή **totalDownloadedBytes**. Για να μην υπερφορτώσουμε τον εξυπηρετητή server, επιτρέπουμε το πολύ 10 νήματα να κατεβάζουν τραγούδια από αυτόν ταυτόχρονα. Στο πρόγραμμά σας επιτρέπεται να χρησιμοποιήσετε για συγχρονισμό μόνο δυαδικούς σηματοφορείς ή/και μεταβλητές συνθήκης. Υποθέστε ότι υπάρχουν δηλωμένες και αρχικοποιημένες οι εξής καθολικές μεταβλητές:

```
char *serverName; char *fileNames[]; int numberOfFiles;
```

### ΑΣΚΗΣΗ 4 (15%)

Εξηγήστε τη λειτουργία του προγράμματος που παρουσιάζεται στον Σχήμα 2. Πόσες γραμμές εκτυπώνονται, και τι περιέχουν; Είναι σωστό το πρόγραμμα ή περιέχει λάθη;