

3. Μια μοναδική εντολή υψηλού επιπέδου είναι **πάντα** ατομική (atomic) σε χαμηλό επίπεδο.

\*

☒ Λάθος



☐ Σωστό

4. Οι κλήσεις συστήματος (system calls), στη γενική περίπτωση, καλούνται απευθείας από τα προγράμματα των χρηστών.

\*

☐ Λάθος



☒ Σωστό



5. Σε συστήματα με μια ΚΜΕ ΔΕΝ μπορούν να δημιουργηθούν συνθήκες ανταγωνισμού (race conditions) όπου η σωστή λειτουργία του κώδικα A εξαρτάται από το αν θα υπάρξει ανταγωνισμός με τον κώδικα B.

\*

☒ Λάθος



☐ Σωστό

6. Σε ποιο τμήμα μνήμης αποθηκεύονται οι τοπικές μεταβλητές μιας συνάρτησης, όταν αυτή κληθεί;

\*

☐ Στο σωρό (heap)

☐ Στο τμήμα που περιέχει τις μεταβλητές που έχουν ρητά αρχικοποιηθεί

☐ Στο τμήμα text που περιέχει τις εντολές του προγράμματος σε γλώσσα μηχανής

☒ Στη στοίβα (stack)



7. Τι εμφανίζει το παρακάτω πρόγραμμα; \*

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

void forkexample() {
    if (fork() == 0)
        printf("Hello from Child! - ");
    else
        printf("Hello from Parent! - ");
}

int main() {
    forkexample();
    return 0;
}
```

☒ Και οι δύο απαντήσεις είναι πιθανές, εξαρτάται από τις αποφάσεις χρονοπρογραμματισμού που θα πάρει το ΛΣ ✓

☐ Hello from Parent! - Hello from Child! -

☐ Hello from Child! - Hello from Parent! -

☐ Hello from Parent! -



8. Γράψτε ψευδοκώδικα, χρησιμοποιώντας μεταβλητές συνθήκης, ώστε να υλοποιείται η παρακάτω λειτουργικότητα. Έστω νήματα τύπου A και B. Κάθε νήμα τύπου A μπορεί να εκτελεστεί όταν η διαμοιραζόμενη μεταβλητή *counter* έχει τιμή μεγαλύτερη από 10, αλλιώς αναστέλλεται καλώντας την *pthread\_cond\_wait()*. Κάθε νήμα B αυξάνει την κοινή μεταβλητή *counter* και όταν αυτή γίνει μεγαλύτερη από 10 ειδοποιεί, καλώντας την *pthread\_cond\_signal()*, κάποιο νήμα τύπου A να προχωρήσει.