

# Λειτουργικά Συστήματα

## 1η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

### ΕΚΦΩΝΗΣΗ

Για την 1η εργαστηριακή άσκηση καλείστε να δημιουργήσετε ένα πρόγραμμα σε C το οποίο θα κάνει τα εξής:

- Δέχεται από την γραμμή εντολών κατά την εκτέλεσή του ένα και μοναδικό όρισμα, το οποίο είναι ένα όνομα αρχείου. Για παράδειγμα, το πρόγραμμα θα πρέπει να εκτελείται ως εξής:

```
$ ./a.out output.txt
```

- Αν το πρόγραμμα κληθεί χωρίς κανένα όρισμα, ή με δύο ή περισσότερα ορίσματα, τότε πρέπει να τυπώνει το εξής μήνυμα σφάλματος (στην έξοδο σφαλμάτων stderr) και να τερματίζει με κωδικό επιστροφής 1:

```
$ ./a.out
```

```
Usage: ./a.out filename
```

```
$ ./a.out file1.txt file2.txt
```

```
Usage: ./a.out filename
```

- Αν το αρχείο που έχει δοθεί στο πρόγραμμα υπάρχει ήδη, τότε το πρόγραμμα τυπώνει το παρακάτω μήνυμα σφάλματος στο stderr και τερματίζει με κωδικό επιστροφής 1:

```
$ touch output.txt
```

```
$ ./a.out output.txt
```

```
Error: output.txt already exists
```

*HINT: Για τον έλεγχο ύπαρξης του αρχείου, δείτε την κλήση συστήματος **stat**.*

- Το πρόγραμμα επίσης πρέπει να μπορεί να κληθεί μαζί με το flag **--help**, στην οποία περίπτωση τυπώνει το παρακάτω μήνυμα στην έξοδο (stdout) και τερματίζει με κωδικό επιστροφής 0:

```
$ ./a.out --help
```

```
Usage: ./a.out filename
```

- Σε κανονική λειτουργία, το πρόγραμμα δημιουργεί μία διεργασία παιδί. Τόσο η διεργασία πατέρα όσο και η διεργασία παιδί πρέπει να γράψουν από ένα μήνυμα σε

αυτό. Συγκεκριμένα, η διεργασία πατέρας θα γράψει το μήνυμα **[PARENT]** **getpid()=\$PID, getppid()=\$PPID**, ενώ η διεργασία παιδί το μήνυμα **[CHILD]** **getpid()=\$PID, getppid()=\$PPID**. Η τελική μορφή του αρχείου θα πρέπει να μοιάζει κάπως έτσι:

```
$ ./a.out output.txt
$ cat output.txt
[CHILD] getpid()=100, getppid()=99
[PARENT] getpid()=99, getppid()=14
```

*HINT: Για την εγγραφή στο αρχείο πρέπει να χρησιμοποιήσετε την κλήση συστήματος **write**. Για την μορφοποίηση του κειμένου, πιθανότατα θα σας φανεί η χρήσιμη η συνάρτηση **snprintf**.*

## ΕΡΩΤΗΣΕΙΣ ΠΡΟΕΤΟΙΜΑΣΙΑΣ

1. Πώς περνάμε ορίσματα σε ένα πρόγραμμα κατά την εκτέλεσή του από την γραμμή εντολών;
2. Πώς γίνεται η δημιουργία διεργασιών παιδιών με την κλήση συστήματος `fork()`;
3. Ποια η διαφορά κλήσης συστήματος και συνάρτησης;
4. Ποια διεργασία θα γράψει πρώτη στο αρχείο στην υλοποίησή σας; Είναι συγκεκριμένη η σειρά, ή απλώς τυχαία;

## ΕΛΕΓΧΟΣ ΕΡΓΑΣΙΑΣ

Για δική σας βοήθεια, έχουμε φτιάξει ένα σύνολο από tests για το πρόγραμμά σας. Η εκτέλεση των ελέγχων γίνεται ως εξής:

```
$ gcc a.c -o a.out
$ PROG=./a.out python test.py
```

**ΣΗΜΕΙΩΣΗ:** Τα tests που αποτυγχάνουν εμφανίζουν και ένα σχετικό μήνυμα λάθους. **Τονίζεται ότι τα tests δίνονται για δική σας βοήθεια στην υλοποίηση της άσκησης. Δεν είναι υποχρεωτικό να περνάνε όλα προκειμένου να εξεταστείτε και να βαθμολογηθείτε. Ωστόσο, δαπανήστε χρόνο για να δείτε ποια tests αποτυγχάνουν και γιατί, και προσπαθήστε να διορθώσετε κατάλληλα το πρόγραμμά σας.**