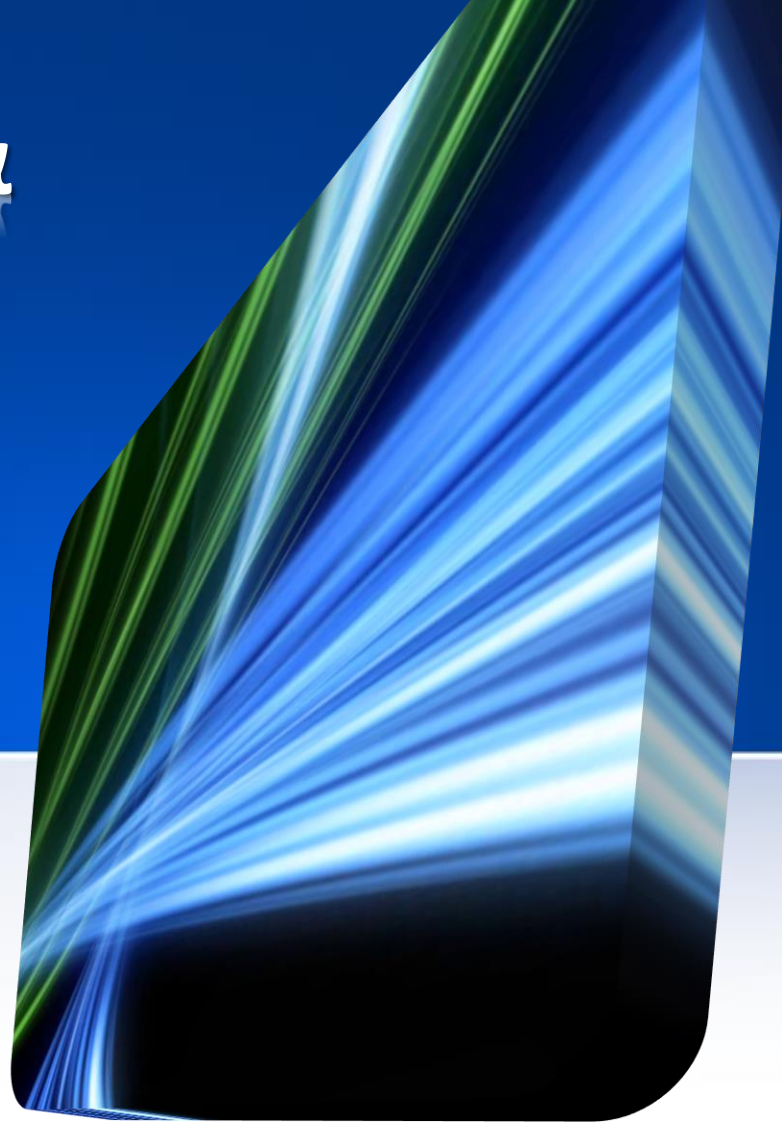
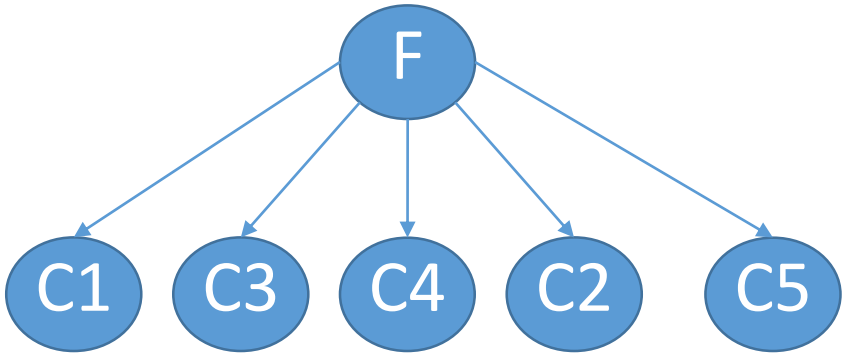
Λειτουργικά Συστήματα
6ο εξάμηνο ΣΗΜΜΥ
Ακ. έτος 2018-2019



Εργαστηριακή Άσκηση 2

Εργαστηριακή Άσκηση 2

Να γραφτεί πρόγραμμα σε γλώσσα προγραμματισμού C και περιβάλλον Linux στο οποίο η διεργασία πατέρας (F) δημιουργεί διεργασίες (C1,C2,C3,C4,C5) σύμφωνα με το παρακάτω δέντρο διεργασιών



Εργαστηριακή Άσκηση 2

Όλες οι διεργασίες παιδιά αρχικά είναι σε κατάσταση pause(), αναμένοντας ένα σήμα.

Η διεργασία-πατέρας τις εκκινεί και στην συνέχεια τις χρονοδρομολογεί με την σειρά που ορίζει ο χρήστης από την γραμμή εντολών.

Π.χ. αν το εκτελέσιμο ονομάζεται assign2, με την παρακάτω εντολή ./assign2 4 5 1 3 2, οι διεργασίες θα ενεργοποιηθούν και στην συνέχεια θα εκτελούνται με την σειρά C4 C5 C1 C3 C2

Εργαστηριακή Άσκηση 2

Η διεργασία πατέρας επιτρέπει σε κάθε διεργασία όταν είναι η σειρά της να εκτελεστεί για 3 sec στην συνέχεια την σταματά και ενεργοποιεί την επόμενη σύμφωνα πάντα με την σειρά που έχει ορίσει ο χρήστης από την γραμμή εντολών. Η διεργασία πατέρας ενεργοποιεί κάθε διεργασία 4 φορές και μετά την τερματίζει.

Εργαστηριακή Άσκηση 2

Κάθε διεργασία παιδί (C1,C2,C3,C4,C5)

εκτυπώνει επαναληπτικά το εξής μήνυμα με καθυστέρηση **ενός δευτερολέπτου**:

Childi Pid is executed (n)

όπου **i** είναι η αρίθμηση της διεργασίας (1 <= **i** <= 5),

Pid είναι το **process id** της διεργασίας

και **n** είναι η αρίθμηση της εκτύπωσης

Εργαστηριακή Άσκηση 2

Προσδοκώμενο αποτέλεσμα:

Οι διεργασίες παιδιά θα χρονοδρομολογηθούν από την διεργασία πατέρας σύμφωνα με τη σειρά που θα δώσει ο χρήστης από γραμμή εντολών. Σε κάθε διεργασία η διεργασία πατέρας θα επιτρέψει να εκτελεστεί 4 φορές για 3 sec και μετά θα την τερματίσει.

Θεωρία Εργ. Άσκησης 2

Ορίσματα Προγράμματος

Η main έχει 2 ορίσματα

- **argc**: Αριθμός ορισμάτων προγράμματος
- **argv**: Πίνακας με τα ορίσματα

```
int main(int argc, char **argv)
```

Θεωρία Εργ. Άσκησης 2

Παράδειγμα1

argv[0]: Το όνομα του προγράμματος

arguments.c	terminal

#include <stdio.h>
int main(int argc, char **argv)
{
  int i;
  for (i=0; i<argc; i++)
  printf("%d \t %s\n", i, argv[i]);
  return 0;
}
	$ gcc arguments.c
	$ ./a.out test1 test2
	0 ./a.out
	1 test1
	2 test2

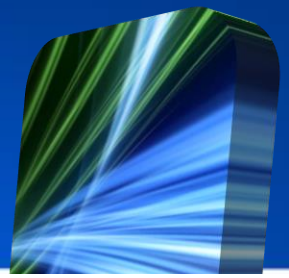
Θεωρία Εργ. Άσκησης 2 (Σήματα)

- Σήμα είναι ένας τρόπος επικοινωνίας των διεργασιών.
- Είναι μια ειδοποίηση που στέλνεται σε μια διεργασία προκειμένου να ενημερωθεί για κάποιο γεγονός.
- Η διεργασία που λαμβάνει ένα σήμα διακόπτει οτιδήποτε κάνει εκείνη την στιγμή, και υποχρεώνεται να χειριστεί το σήμα άμεσα.

Θεωρία Εργ. Άσκησης 2 (Σήματα)

- Κάθε σήμα έχει έναν ακέραιο αριθμό που το αντιπροσωπεύει (1,2 …), καθώς επίσης και ένα συμβολικό όνομα που καθορίζεται συνήθως στο αρχείο /usr/include/signal.h
- Με την εντολή "kill -l" μπορείτε να δείτε έναν κατάλογο σημάτων που υποστηρίζονται από το σύστημά σας

Θεωρία Εργ. Άσκησης 2 (Σήματα)



Signal	Value	Action	Comment
SIGHUP	1	Term	Hangup detected on controlling terminal or death of controlling process
SIGINT	2	Term	Interrupt from keyboard
SIGQUIT	3	Core	Quit from keyboard
SIGILL	4	Core	Illegal Instruction
SIGABRT	6	Core	Abort signal from abort(3)
SIGFPE	8	Core	Floating point exception
SIGKILL	9	Term	Kill signal
SIGSEGV	11	Core	Invalid memory reference
SIGPIPE	13	Term	Broken pipe: write to pipe with no readers
SIGALRM	14	Term	Timer signal from alarm(2)
SIGTERM	15	Term	Termination signal
SIGUSR1	30,10,16	Term	User-defined signal 1
SIGUSR2	31,12,17	Term	User-defined signal 2
SIGCHLD	20,17,18	Ign	Child stopped or terminated
SIGCONT	19,18,25	Cont	Continue if stopped
SIGSTOP	17,19,23	Stop	Stop process
SIGTSTP	18,20,24	Stop	Stop typed at terminal
SIGTTIN	21,21,26	Stop	Terminal input for background process
SIGTTOU	22,22,27	Stop	Terminal output for background process

Θεωρία Εργ. Άσκησης 2 (Σήματα)

Κάθε σήμα μπορεί να έχει έναν χειριστή σημάτων, το οποίο είναι μια συνάρτηση που καλείται όταν λαμβάνει η διεργασία το συγκεκριμένο σήμα. Η συνάρτηση καλείται ασύγχρονα. Όταν στέλνεται το σήμα σε μια διεργασία, το ΛΣ σταματά την εκτέλεση της διεργασίας, και "την αναγκάζει" να καλέσει τη συνάρτηση χειρισμού του σήματος. Όταν εκείνη η συνάρτηση ολοκληρώσει την εκτέλεσή της, η διεργασία συνεχίζει την εκτέλεση από το σημείο που βρισκόταν αμέσως πριν να παραληφθεί το σήμα

Θεωρία Εργ. Άσκησης 2 (Σήματα)

Η κλήση **kill(pid, SIGINT)**

στέλνει το σήμα SIGINTστη διεργασία με PID pid.

Η κλήση **signal(SIGINT, handler)** καθορίζει ότι όταν η διεργασία λάβει σήμα **SIGINT** θα εκτελέσει τη συνάρτηση χειρισμού σήματος **handler**.

Θεωρία Εργ. Άσκησης 2 (pause)

Η κλήση συστήματος **pause()** αναγκάζει τη διεργασία να σταματήσει την εκτέλεση της, έως ότου ληφθεί ένα σήμα

Χρήσιμα links

http://man7.org/linux/man-pages/man7/signal.7.html