

מעבדה 4. קריאות מערכת הפעלה לניהול תהליכים

fork דוגמא של

הסבירו את תוצאות ההרצה של כל דוגמא

```
/* forkfile0.c - first use of fork() */
```

```
#include <stdio.h>
```

```
int main(){
    printf("Before fork:\n");
    fork();
    printf("After fork.\n");
}
```

```
/* forkfile1.c */
```

```
#include <stdio.h>
```

```
int main(){
    printf("Before fork: My pid = %d\n", getpid());
    fork();
    printf("After fork: My pid = %d\n", getpid());
}
```

```
/* forkfile2.c */
```

```
#include <stdio.h>
```

```
int main(){
    int pid;
    pid = getpid();
    printf("Before fork: My pid = %d\n", getpid());
    fork();
    printf("After fork: My pid = %d, pid = %d\n", getpid(), pid);
}
```

```
/* forkfile3.c */
```

```
#include <stdio.h>
```

```
int main(){
    int pid1, pid2;
    pid1 = getpid();
    printf("Before fork: My pid = %d\n", getpid());
    pid2 = fork();
}
```

```

if ( pid2 != 0)
    printf("I am the parent! getpid=%d, pid=%d, id in fork=%d\n",
                                                getpid(), pid1, pid2);
else
    printf("I am the child! getpid=%d, pid=%d, id in fork=%d\n",
                                                getpid(), pid1, pid2);
}

```

תכניות דוגמא של exec.., wait

תחילה, צרו תיקייה `/home/braude/Documents/My`
 בתיקייה זאת, כתבו בקובץ `Hello.c` תכנית בשפת C המדפיס למסך `"Hello world"`
 קמפלו את התכנית `Hello.c` ליצירת קובץ הרצה `Hello.out`

הסבירו את תוצאות ההרצה של כל דוגמא

/ Lab4_2.c - call system date routine using execl */*

```

#include <stdio.h>
#include<unistd.h>

void main(){
    printf("Here comes the date:\n");
    execl("/bin/date", "date", 0);
    printf("ERROR - execl failed.\n");
}

```

/ Lab4_3.c */*

```

#include <stdio.h>
#include<unistd.h>

void main(){
    printf("Here comes the date:\n");
    fork();
    execl("/bin/date", "date", 0);
    printf("That was the date.\n");
}

```

/ Lab4_4.c */*

```

#include <stdio.h>
#include<unistd.h>

void main(){
    int id;
    printf("Here comes the date:\n");
}

```

```

if ( ( id = fork() ) == 0 ) {
    printf("PID is %d and ID is %d.\n", getpid(), id);
    execl("/bin/date", "date",0);
}
printf("PID is %d and ID is %d.\n", getpid(), id);
printf("That was the date.\n");
}

```

/ Lab4_5.c - Use of fork() and wait() */*

```

#include <stdio.h>
#include<unistd.h>

void main(){
    int id, wid, status;
    printf("Here comes the date:\n");
    if ( ( id = fork() ) == 0 ) {
        printf("PID is %d and ID is %d.\n", getpid(), id);
        execl("/bin/date", "date",0);
    }
    wid = wait(&status);
    printf("That was the date.\n");
    printf("PID is %d and ID is %d.\n", getpid(), id);
    printf("wid = %d and status = %d\n", wid, status);
}

```

/ Lab4_6.c - Use of fork() and wait() */*

```

#include <stdio.h>
#include<unistd.h>
void main(){
    int id, wid, status;
    printf("Here comes the date:\n");
    switch ( id = fork() ) {
        case -1:
            perror("fork");
            exit(1);
        case 0:
            execl("/bin/date", "date",0);
            perror("execl");
            exit(1);
        default:
            break;
    } /* switch */
    wid = wait(&status);
}

```

```
printf("That was the date.\n");
}
```

```
/* Lab4_7.c - Use of fork() and execv() */
```

```
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
void main(){
    int id;
    printf("PID Before fork: %d\n", getpid());
    if((id=fork())==0){
        printf("After fork:\n");
        printf("PID is %d and ID is %d.\n", getpid(), id);
        execv("/home/braude/Documents/My/Hello.out", NULL);
        printf("Child's exit.\n");
    }
    else{
        printf("After fork:\n");
        printf("PID is %d and ID is %d.\n", getpid(), id);
        printf("Parent's exit.\n");
    }
}
```

תרגיל 1 – System Calls For Process

עלייך לכתוב תכנית בשפת C תחת מ"ה לינוקס בשם runInLoop.c אשר מריצה בלולאה תכנית אחרת. התוכנית תקבל משורת הפקודה שם תוכנית, Prog ומספר N ותריץ את התוכנית Prog בלולאה N פעמים. בסוף ההרצה, יודפס למסך End. לדוגמא, אם קיימת בתיקייה תכנית בשם hello.c:

```
#include <stdio.h>

int main(){
    printf("Hello\n");
}
```

ובוצע: `cc hello.c -o hello`

בהרצת התוכנית שכתבת עם הפרמטרים hello ו 5, פלט התוכנית שלך נדרש להיות:

```
$ runInLoop hello 5
```

```
Hello
```

```
Hello
```

```
Hello
```

```
Hello
```

```
Hello
```

```
END
```

תרגיל 2 :

תחת מערכת ההפעלה UNIX, כתבו תכנית בשפת C אשר מנהלת shell מצומצם רק עבור פקודות ללא פרמטרים (כגון: date, ps, ls). התכנית תחילה תציג סימן \$ ותחכה לקלט מהמשתמש. לכל קלט, התוכנית תתייחס כפקודה להרצה ותציג את הפלט המתאים. עבור הקלט q, התכנית תסתיים.

עבור קלט שאיננו פקודה חוקית, התוכנית תדפיס למסך ERROR ERROR ERROR.

שם התוכנית lab4tar2.c. יש להשתמש בקריאות מ"ה fork, wait, exec.

דוגמאת הרצה לתרגיל 2:

```
./lab4tar2.out
```

```
$ps
```

PID	TTY	TIME	CMD
3938	pts/4	00:00:00	tcsh
3948	pts/4	00:00:00	lab4tar2.out
3988	pts/4	00:00:00	ps

```
$date
```

```
Sun Mar 29 16:00:15 IST 2020
```

```
$xxxx
```

```
ERROR ERROR ERROR
```

```
$q
```

```
END
```

****קלט מהמשתמש ב אדום.**

הוראות הגשה:

יש להגיש קובץ WORD ובו:

1. שמות ות.ז. של שני המגישים.
 2. תכנית פתרון לתרגיל 1 וצילום מסך הרצה.
 3. תכנית פתרון לתרגיל 2 וצילום מסך הרצה.
- בצילומי המסך יש להריץ את התוכניות על פי הדוגמאות הנתונות.

ההגשה במודל. יש להגיש את הקובץ על ידי שני בני הזוג.

בהצלחה!!!