

Assignment 2:

Unit Processes

Duc M. Le

Thompson Rivers University

COMP 3411 – Operating System

Kevin O'Neil

September 21, 2021

Question 1

Program testOS.c

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

int main (int args, char *argv[])
{
    pid_t fork_return;
    pid_t pid;
    pid=getpid();
    fork_return = fork();
    if (fork_return==-1) {
        // When fork() returns -1, an error has happened.
        printf("\nError creating process " );
        return 0;
    }
    if (fork_return==0) {
        // When fork() returns 0, we are in the child process.
        printf("\n\nThe values are Child ID = %d, Parent ID=%d\n", getpid(), getppid(
    ));
        execl("/bin/cat", "cat", "-b", "-v", "-t", argv[1], NULL);
        printf("This line is printed after exec* call\n");
    }
    else {
        // When fork() returns a positive number, we are in the parent process
        // and the return value is the PID of the newly created child process.
        wait(NULL);
        printf("\nChild Completes " );
        printf("\nIn the Parent Process\n");
        printf("Child Id = %d, Parent ID = %d\n", getpid(), getppid());
    }
    return 0;
}
```

Input file: test.txt

```
This is line one in my text file.
This is line two in my text file.
This is line three in my text file.
Hello there
It's over Anakin. I have the high ground!
You underestimate my power!
Unlimited power!
Linux provides many commands for manipulating text!
```

Screen capture for Q1

```
Script started on Mon 20 Sep 2021 11:55:26 PM PDT
[]0;kocomp341130@cs2:~/comp3411/Assignments/A2[] [?1034h[kocomp341130@cs2
A2]$ ./a.out test.txt
```

```
The values are Child ID = 42119, Parent ID=42118
  1      This is line one in my text file.^M
  2      This is line two in my text file.^M
  3      This is line three in my text file.^M
  4      Hello there^M
  5      It's over Anakin. I have the high ground!^M
  6      You underestimate my power!^M
  7      Unlimited power!^M
  8      Linux provides many commands for manipulating text!^M
```

Child Completes

In the Parent Process

Child Id = 42118, Parent ID = 42099

```
[]0;kocomp341130@cs2:~/comp3411/Assignments/A2[] [kocomp341130@cs2 A2]$ exit
exit
```

Script done on Mon 20 Sep 2021 11:56:03 PM PDT

Question 2.

- a. The program did not print a message after the `exec*` call as the `printf` command belongs to the parent program. By the time `exec*` was called, the current process image was already being replaced by the new process image, which started executing the new program and would not return control unless an error had occurred.
- b. The parent program is the whole compiled `testOS.c` executable `a.out` itself.
- c. I would just comment out the `exec*` call, the child would then be an exact copy of the parent and run concurrently along with it, though the `cat` command would not be executed.

Screen capture for Q2

Script started on Mon 20 Sep 2021 11:56:32 PM PDT

```
]0;kocomp341130@cs2:~/comp3411/Assignments/A2    [?1034h[kocomp341130@cs2 A2]$ vi
test0St    [K.tx    [K    [Kc
[?1049h [?1h = [2;1H??[6n [2;1H
[1;1H [1;30r [?12;25h [?121 [?25h [27m [m [H [2J [?251 [30;1H"testOS.c" [dos]
31L, 971C [>c [1;1H [35m#include [m [31m<stdio.h> [m
[35m#include [m [31m<sys/types.h> [m
[35m#include [m [31m<unistd.h> [m

[32mint [m main ( [32mint [m args, [32mchar [m *argv[])
{
    pid_t fork_return;
    pid_t pid;
    pid=getpid();
    fork_return = fork();
    [38;5;130mif [m (fork_return== - [31m1 [m) {
        [34m// When fork() returns -1, an error has happened. [m
        printf( [31m" [m [35m\n [m [31mError creating process " [m );
        [38;5;130mreturn [m [31m0 [m;
    }

    [38;5;130mif [m (fork_return== [31m0 [m) {
        [34m// When fork() returns 0, we are in the child process. [m
        printf( [31m" [m [35m\n\n [m [31mThe values are Child ID =
[m [35m%d [m [31m, Parent ID= [m [35m%d\n [m [31m" [m, getpid(), getppid());
        execl( [31m"/bin/cat" [m, [31m"cat" [m, [31m"-b" [m, [31m"-v" [m, [31m"-
t" [m, argv[ [31m1 [m], [31mNULL [m];
        printf( [31m"This line is printed after exec* call [m [35m\n [m [31m" [m];
    }

    [38;5;130melse [m {
        [34m// When fork() returns a positive number, we are in the parent process
[m [34m// and the return value is the PID of the newly created child process. [m
```

```

    wait( [31mNULL [m];
    printf( [31m" [m [35m\n [m [31mChild Completes " [m );
    printf( [31m" [m [35m\n [m [31mIn the Parent
Process [m [35m\n [m [31m" [m);
    printf( [31m"Child Id = [m [35m%d [m [31m, Parent ID =
[m [35m%d\n [m [31m" [m, getpid(), getppid());
} [30;103H19,5 [10CTop [19;5H [?121 [?25h [?251 [30;1H [1m-- INSERT --
[m [30;13H [K [30;103H19,5 [10CTop [19;5H [?121 [?25h [?251/execl( [31m"/bin/c
at" [m, [31m"cat" [m, [31m"-b" [m, [31m"-v" [m, [31m"-t" [m,
argv[ [31ml [m],
[31mNULL [m]; [30;106H6 [19;6H [?121 [?25h [?251 [34m//execl("/bin/cat",
"cat", "-b", "-v", "-t", argv[1],
NULL); [m [30;106H7 [19;7H [?121 [?25h [?251 [30;1H [K [30;103H19,6 [10CTop [1
9;6H [?121 [?25h [?251 [30;103H [K [30;1H: [?121 [?25hq [?251 [?121 [?25h [?2
51 [30;2H [K [30;2H [?121 [?25hx [?251 [?121 [?25h [?251"testOS.c" [dos] 31L,
973C written
[?11 > [?121 [?25h [?10491 ]0;kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp3
41130@cs2 A2]$ vi testOS.c [7Pexit ./a.out
test.txt [10Plogout [1Pps -
f clear ps -f ./a.out & [4Pps -
f clear ps -f logout ./a.out
test.txt exit [K vi
testOS.c [Kgcc testOS.c
]0;kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp341130@cs2 A2]$ gcc
testOS.c [1Pvi [C [C [C [C [C [C [C [C
[7Pexit ./a.out test.txt

```

The values are Child ID = 42150, Parent ID=42149
This line is printed after exec* call

Child Completes

In the Parent Process

Child Id = 42149, Parent ID = 42124

```

]0;kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp341130@cs2 A2]$ exit
exit

```

Script done on Mon 20 Sep 2021 11:57:24 PM PDT

Script started on Tue 21 Sep 2021 12:21:14 AM PDT

```
[?]0; kocomp341130@cs2:~/comp3411/Assignments/A2 [?]1034h[kocomp341130@cs2 A2]$ exit ps -f kill -9 42313 4 [8Pps -f kill 42313 4 [1P1776 [1P 1776 [3Pps -f kill 1776 176 [4Pps -f ./a.out & [5Pexit ./a.out & [1] 42358 [?]0; kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp341130@cs2 A2]$
```

The values are Child ID = 42359, Parent ID=42358

ps -f

| UID | PID | PPID | C | STIME | TTY | TIME | CMD |
|------|-------|-------|---|-------|--------|----------|--------------|
| 1776 | 42341 | 42340 | 0 | 00:21 | pts/20 | 00:00:00 | bash -i |
| 1776 | 42358 | 42341 | 0 | 00:21 | pts/20 | 00:00:00 | ./a.out |
| 1776 | 42359 | 42358 | 0 | 00:21 | pts/20 | 00:00:00 | cat -b -v -t |
| 1776 | 42360 | 42341 | 0 | 00:21 | pts/20 | 00:00:00 | ps -f |

```
[1]+  Stopped                  ./a.out
[kocomp341130@cs2:~/comp3411/Assignments/A2] [cocomp341130@cs2 A2]$ ps -
f      ./a.out &      [5Pexit      ./a.out
&      [C      [Ktest.xtx      [K      [K      [Ktxt
```

The values are Child ID = 42362, Parent ID=42361

```
1 This is line one in my text file.^M
2 This is line two in my text file.^M
3 This is line three in my text file.^M
4 Hello there^M
5 It's over Anakin. I have the high ground!^M
6 You underestimate my power!^M
7 Unlimited power!^M
8 Linux provides many commands for manipulating text!^M
```

Child Completes

In the Parent Process

Child Id = 42361, Parent ID = 42341

```

[kocomp341130@cs2:~/comp3411/Assignments/A2] [cocomp341130@cs2 A2]$ ps -
f

```

| UID | PID | PPID | C | STIME | TTY | TIME | CMD |
|------|-------|-------|---|-------|--------|----------|--------------|
| 1776 | 42341 | 42340 | 0 | 00:21 | pts/20 | 00:00:00 | bash -i |
| 1776 | 42358 | 42341 | 0 | 00:21 | pts/20 | 00:00:00 | ./a.out |
| 1776 | 42359 | 42358 | 0 | 00:21 | pts/20 | 00:00:00 | cat -b -v -t |
| 1776 | 42363 | 42341 | 0 | 00:21 | pts/20 | 00:00:00 | ps -f |

```
l] 0;kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp341130@cs2 A2]$ kill
-9 42358
```

```

[kocomp341130@cs2:~/comp3411/Assignments/A2] [kocomp341130@cs2 A2]$ ps -
f

```

```

UID          PID    PPID    C  STIME TTY          TIME CMD
1776        42341  42340    0 00:21 pts/20      00:00:00 bash -i
1776        42366  42341    0 00:22 pts/20      00:00:00 ps -f
[1]+  Killed                  ./a.out

```

```
]0;kocomp341130@cs2:~/comp3411/Assignments/A2 [kocomp341130@cs2 A2]$ exit  
exit
```

Script done on Tue 21 Sep 2021 12:22:13 AM PDT