

Assignment 1 - Question 4

Author - Ashutosh Sahu - 11840260

Note - In case the code/text formatting looks odd in this PDF, you can view the original Dropbox Paper [here](#).

Write a simple program in your favorite language and `strace` its execution on bash.

This is the code on which I will be running `strace` on:

```
#include<sys/wait.h>
#include<unistd.h>
#include<stdlib.h>
#include<stdio.h>
#include <sched.h>

int fn(void *arg)
{
    int var;
    printf("\nHi There! I am the clone process. \nI don't
really want to create any mischief like the previous child. So
I will just exit as soon as you press Enter.\n\n\nPress Enter
to continue.\n");
    getchar();
    return 0;
}

int main(int argc, char *argv[])
{
    char* c;
    FILE *fptr;
```

```

    void *pchild_stack = malloc(1024 * 1024);

    printf("Hello! I am the parent process. My PID is %d\n
You can attach me to strace now. Press Enter when you are don
e.", (int) getpid());
    getchar();
    int rc = fork();
    if (rc < 0)
    {
        // fork failed; exit
        fprintf(stderr, "fork failed\n");
        exit(1);
    }
    else if (rc == 0)
    {
        // child (new process)
        printf("\nHello, I am the child. My PID is %d
\n", (int) getpid());
        sleep(3);
        printf("\nWriting to file '123temp.txt'. This file wil
l be deleted before my parent process terminates.\n");
        fptr = fopen("123temp.txt", "w");
        fprintf(fptr, "Hi there! This is just a test content. I
f this file exists for more than 1 second, it means that there
was some problem with the executing process.");
        sleep(6);
        fclose(fptr);
        printf("\nWritten. Now I will exit! Bye!\n");
    }
    else
    {
        // parent goes down this path (main)

```

```

        int rc_wait = wait(NULL);
        sleep(5);
        printf("\nI am the parent process. Looks like
my child has created some garbage file into your computer. I a
m reading the file contents now.\n");
        fptr = fopen("123temp.txt", "r");
        fscanf(fptr, c);
        fclose(fptr);
        sleep(8);
        printf("\nRead. I will be deleting what my mis
chievous child created.\n");
        remove("123temp.txt");
        sleep(5);
        printf("\nDeleted. Now I will be creating a cl
one process.\n");
        sleep(4);
        if (pchild_stack == NULL )
        {
            printf("ERROR: Unable to allocate me
mory for cloning.\n");
            exit(EXIT_FAILURE);
        }

        int pid = clone(fn, pchild_stack + (1024 * 102
4), SIGCHLD, argv[1]);

        if ( pid < 0 )
        {
            printf("ERROR: Unable to create the ch
ild process for cloning.\n");
            exit(EXIT_FAILURE);

```

```

        }
        wait(NULL);
        free(pchild_stack);
        printf("\nLooks like clone child process has e
xited. Now I'll exit too. Have a great day!\n");
    }
    return 0;
}

```

Running `strace` on it will generate the following output:

```

ashutosh@ashutosh-OMEN:~$ sudo strace -f -p 7897 -e trace=read,write,fork,clone
strace: Process 7897 attached
read(0, "\n", 1024)          = 1
clone(strace: Process 7930 attached
child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7f0eebe46810) = 7930
[pid 7930] write(1, "\n", 1)          = 1
[pid 7930] write(1, "Hello, I am the child. My PID is"... , 38) = 38
[pid 7930] write(1, "\n", 1)          = 1
[pid 7930] write(1, "Writing to file '123temp.txt'. T"... , 94) = 94
[pid 7930] write(3, "Hi there! This is just a test co"... , 147) = 147
[pid 7930] write(1, "\n", 1)          = 1
[pid 7930] write(1, "Written. Now I will exit! Bye!\n", 31) = 31
[pid 7930] +++ exited with 0 +++
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7930, si_uid=1000, si_status=0, si_etime=0, si_stime=0} ---
write(1, "\n", 1)          = 1
write(1, "I am the parent process. Looks l"... , 131) = 131
write(1, "\n", 1)          = 1
write(1, "Read. I will be deleting what my"... , 60) = 60
write(1, "\n", 1)          = 1
write(1, "Deleted. Now I will be creating "... , 49) = 49
clone(strace: Process 7990 attached
<unfinished ...>
[pid 7990] write(1, "\nHi There! I am the clone proces"... , 154 <unfinished ...>
[pid 7897] <... clone resumed> child_stack=0x7f0eebc53000, flags=SIGCHLD) = 7990
[pid 7990] <... write resumed> )          = 154
[pid 7990] write(1, "Press Enter to continue.\n", 25) = 25
[pid 7990] read(0, "\n", 1024)          = 1
[pid 7990] +++ exited with 0 +++
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7990, si_uid=1000, si_status=0, si_etime=0, si_stime=0} ---
write(1, "\n", 1)          = 1
write(1, "Looks like clone child process h"... , 80) = 80
+++ exited with 0 +++

```

I have tried to make this program as much friendly as possible. So it is self explanatory about at which point, which action is taking place. All of these actions namely read, write, fork and clone mentioned in the program are traced by `strace`.