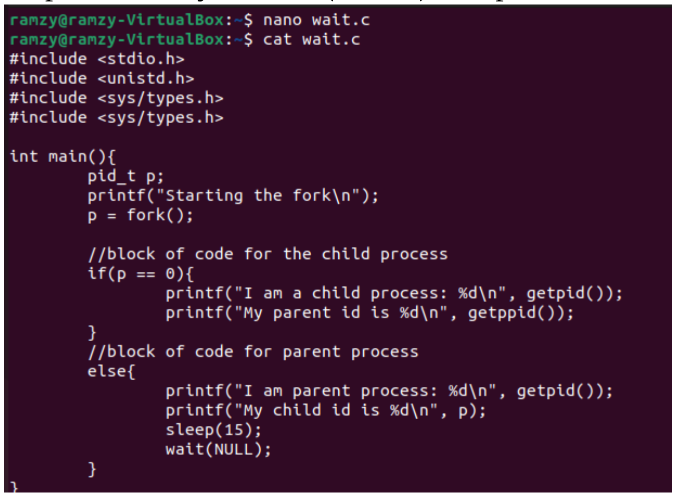
Name : Ramzy Izza Wardhana
NIM : 21/472698/PA/20322
Class : IUP CS B

Activity 4.2 – Process & Process Management

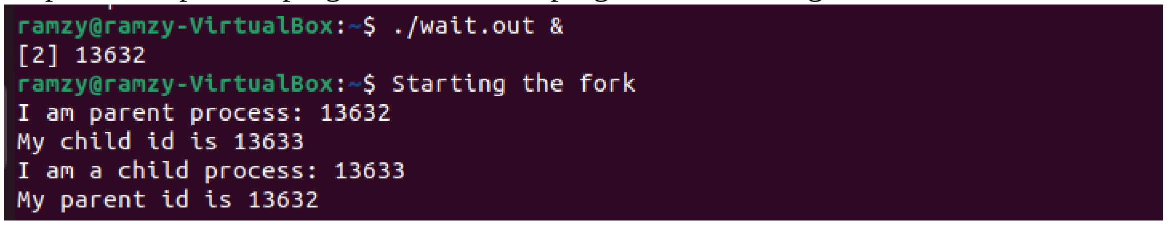
1. Modify the wait.c by putting the wait() function after the sleep() function. Compile and run the result in the background. Take some screenshots. 2. Take a screenshot of the process table. What happens with the child process?

   Step 1 – Modify the wait(NULL) to replace it after the sleep() function by using nano



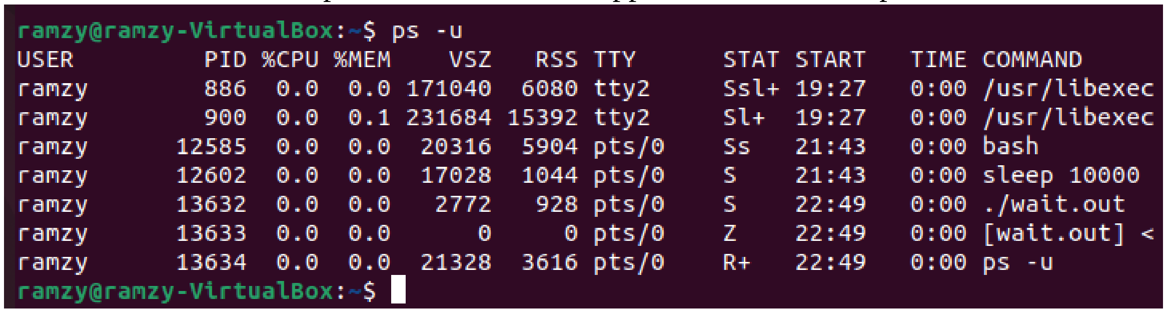
```
ramzy@ramzy-VirtualBox:~$ nano wait.c
ramzy@ramzy-VirtualBox:~$ cat wait.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/types.h>

int main(){
        pid_t p;
        printf("Starting the fork\n");
        p = fork();

        //block of code for the child process
        if(p == 0){
                printf("I am a child process: %d\n", getpid());
                printf("My parent id is %d\n", getppid());
        }
        //block of code for parent process
        else{
                printf("I am parent process: %d\n", getpid());
                printf("My child id is %d\n", p);
                sleep(15);
                wait(NULL);
        }
```

   Step 2 – Compile the program and run the program in the background



```
ramzy@ramzy-VirtualBox:~$ ./wait.out &
[2] 13632
ramzy@ramzy-VirtualBox:~$ Starting the fork
I am parent process: 13632
My child id is 13633
I am a child process: 13633
My parent id is 13632
```

2. Take a screenshot of the process table. What happens with the child process?



```
ramzy@ramzy-VirtualBox:~$ ps -u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
ramzy        886  0.0  0.0 171040  6080 tty2     Ssl+ 19:27   0:00 /usr/libexec
ramzy        900  0.0  0.1 231684 15392 tty2     Sl+  19:27   0:00 /usr/libexec
ramzy      12585  0.0  0.0  20316  5904 pts/0    Ss   21:43   0:00 bash
ramzy      12602  0.0  0.0  17028  1044 pts/0    S    21:43   0:00 sleep 10000
ramzy      13632  0.0  0.0   2772   928 pts/0    S    22:49   0:00 ./wait.out
ramzy      13633  0.0  0.0      0     0 pts/0    Z    22:49   0:00 [wait.out] <
ramzy      13634  0.0  0.0  21328  3616 pts/0    R+   22:49   0:00 ps -u
ramzy@ramzy-VirtualBox:~$
```

   Taking a closer look at the process table above, the child process (13633) is still in the (Z) Zombie state while the parent process (13632) is in the (S) interruptible sleep state. This happens since the parent's process took a longer process and the program works in a procedural way in which it reads the wait() function right after the sleep(). Making the Zombie state occurs during the execution where the parent process has not finished, thus zombie state that appears from the child would not be clear yet.