

Tugas Praktikum SKJ Ke-4

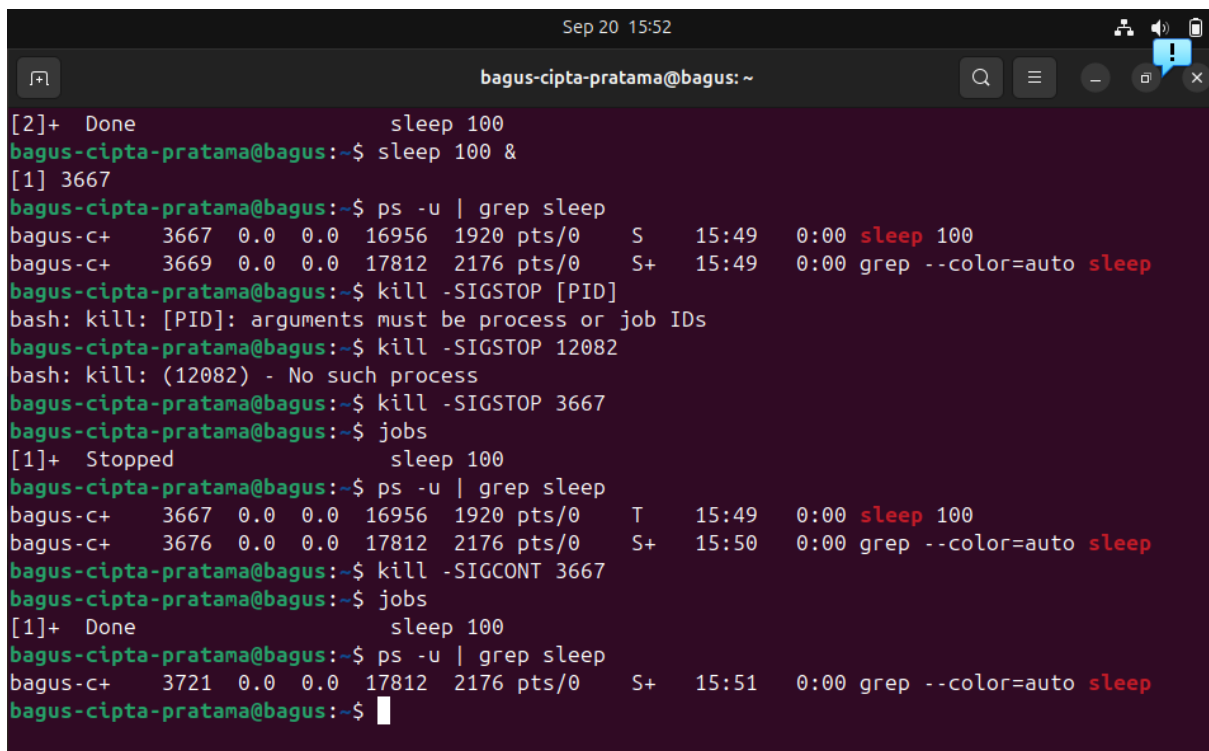
Nama : Bagus Cipta Pratama

NIM : 23/516539/PA/22097

Kelas : KOMC

Activity 4.1 :

1. Create a sleep process and send a SIGSTOP signal to put this process into stopped state. Take a screenshot that shows that this process is now in the stopped state (T). Hint: Use --help to see how to use the kill command.
2. Send a SIGCONT signal to the stopped sleep process. Take a screenshot that shows that this process is no longer in the stopped state.



```
Sep 20 15:52
bagus-cipta-pratama@bagus: ~
[2]+  Done                  sleep 100
bagus-cipta-pratama@bagus:~$ sleep 100 &
[1] 3667
bagus-cipta-pratama@bagus:~$ ps -u | grep sleep
bagus-c+  3667  0.0  0.0 16956 1920 pts/0    S   15:49   0:00  sleep 100
bagus-c+  3669  0.0  0.0 17812 2176 pts/0    S+  15:49   0:00  grep --color=auto  sleep
bagus-cipta-pratama@bagus:~$ kill -SIGSTOP [PID]
bash: kill: [PID]: arguments must be process or job IDs
bagus-cipta-pratama@bagus:~$ kill -SIGSTOP 12082
bash: kill: (12082) - No such process
bagus-cipta-pratama@bagus:~$ kill -SIGSTOP 3667
bagus-cipta-pratama@bagus:~$ jobs
[1]+  Stopped                  sleep 100
bagus-cipta-pratama@bagus:~$ ps -u | grep sleep
bagus-c+  3667  0.0  0.0 16956 1920 pts/0    T   15:49   0:00  sleep 100
bagus-c+  3676  0.0  0.0 17812 2176 pts/0    S+  15:50   0:00  grep --color=auto  sleep
bagus-cipta-pratama@bagus:~$ kill -SIGCONT 3667
bagus-cipta-pratama@bagus:~$ jobs
[1]+  Done                  sleep 100
bagus-cipta-pratama@bagus:~$ ps -u | grep sleep
bagus-c+  3721  0.0  0.0 17812 2176 pts/0    S+  15:51   0:00  grep --color=auto  sleep
bagus-cipta-pratama@bagus:~$
```

Penjelasan kode :

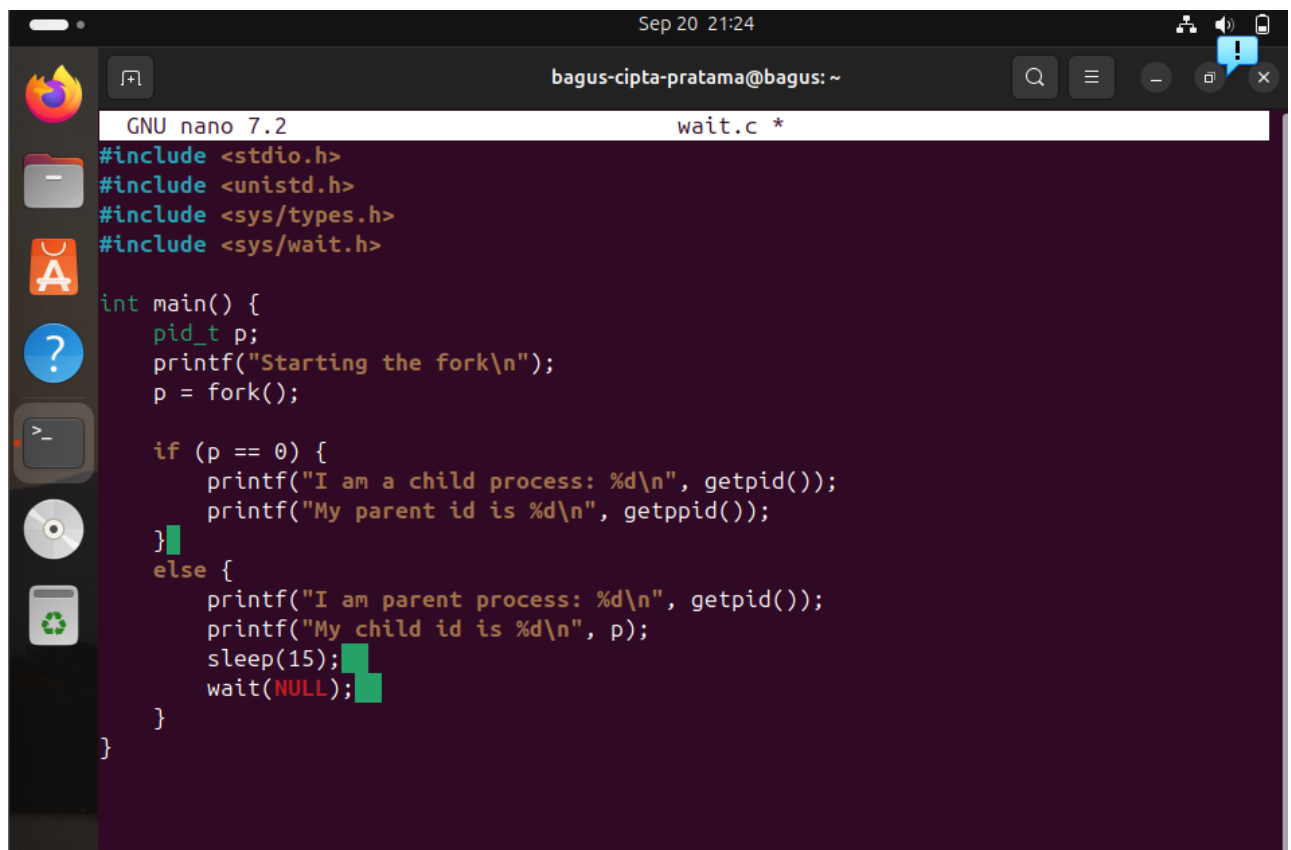
Pertama , kita menjalankan perintah untuk membuat proses sleep yang berjalan selama 100 detik di latar belakang . lalu setelah itu kita dapat melihat bahwa sleep 100 sekarang berjalan di latar belakang dengan Process ID 3667 . setelah itu , kita menggunakan perintah ps untuk melihat daftar proses yang sedang berjalan dan menemukan PID dari proses sleep . output akan menunjukkan proses sleep dengan PID yang teridentifikasi . selanjutnya , untuk menghentikan proses sleep , kita kirim sinyaln sigstop dengan nomor PID dari proses sleep . kemudian kita mengecek bahwa proses

tersebut sekarang berada dalam keadaan ‘stopped’ dengan menjalankan perintah jobs .

Pada tahap kedua , kita akan mengirimkan sinyal sigcont untuk melanjutkan proses sleep dan di akhir , kita bisa melihat bahwa proses telah Kembali dengan menjalankan ps .

Activity 4.2 :

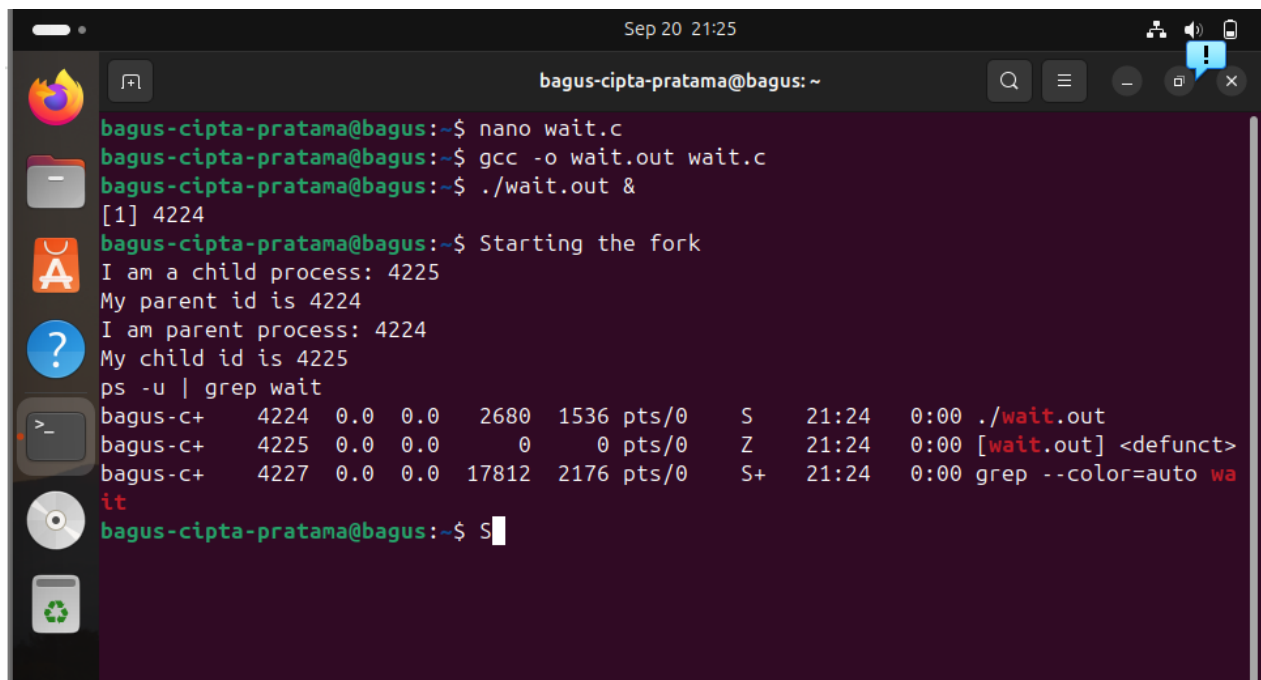
1. Modify the wait.c by putting the wait() function after the sleep() function. Compile and run the result in the background. Take some screenshots.
2. Take a screenshot of the process table. What happens with the child process?



```
GNU nano 7.2          wait.c *
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

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

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



```
bagus-cipta-pratama@bagus: ~  
bagus-cipta-pratama@bagus:~$ nano wait.c  
bagus-cipta-pratama@bagus:~$ gcc -o wait.out wait.c  
bagus-cipta-pratama@bagus:~$ ./wait.out &  
[1] 4224  
bagus-cipta-pratama@bagus:~$ Starting the fork  
I am a child process: 4225  
My parent id is 4224  
I am parent process: 4224  
My child id is 4225  
ps -u | grep wait  
bagus-c+  4224  0.0  0.0  2680  1536 pts/0    S   21:24   0:00  ./wait.out  
bagus-c+  4225  0.0  0.0      0      0 pts/0    Z   21:24   0:00  [wait.out] <defunct>  
bagus-c+  4227  0.0  0.0 17812  2176 pts/0    S+  21:24   0:00  grep --color=auto wa  
it  
bagus-cipta-pratama@bagus:~$
```

Penjelasan Kode :

Hal pertama yang kita lakukan disini adalah memodifikasi file C source code wait.c dengan mengubah urutan fungsi . kita membuka file wait.c dengan editor teks nano dan mengubah isi filenya sehingga fungsi wait dipanggil setelah fungsi sleep(15) . setelah melakukan modifikasi , simpan dan keluar dari editor .

Selanjutnya , kita kompilasi kode program menggunakan gcc dan menjalankan wait.out di latar belakang . terlihat output yang keluar dari proses tersebut pada gambar diatas . setelah itu kita menggunakan ps-u untuk melihat daftar proses dan memastikan bahwa proses anak masih ada dalam table proses . dari screenshot diatas diperlihatkan bahwa proses induk sedang tidur selama 15 detik sementara proses anak sedang menunggu .

Karena proses induk tidur selama 15 detik, proses anak akan selesai lebih cepat dan berubah menjadi zombie process sementara menunggu proses induk membaca status keluarannya. Setelah proses induk selesai "tidur" dan memanggil fungsi wait(), status proses anak akan dibersihkan dari tabel proses. Pada titik ini, proses anak tidak lagi terlihat dalam tabel proses, dan zombie process akan

hilang. Jika kita menjalankan perintah `ps -u` sebelum fungsi `wait()` dipanggil, kita mungkin akan melihat proses anak dalam status "defunct" atau zombie . Setelah fungsi `wait()` dipanggil oleh proses induk, proses anak akan dibersihkan dari tabel proses, dan kita tidak akan lagi melihatnya dalam hasil `ps -u`.