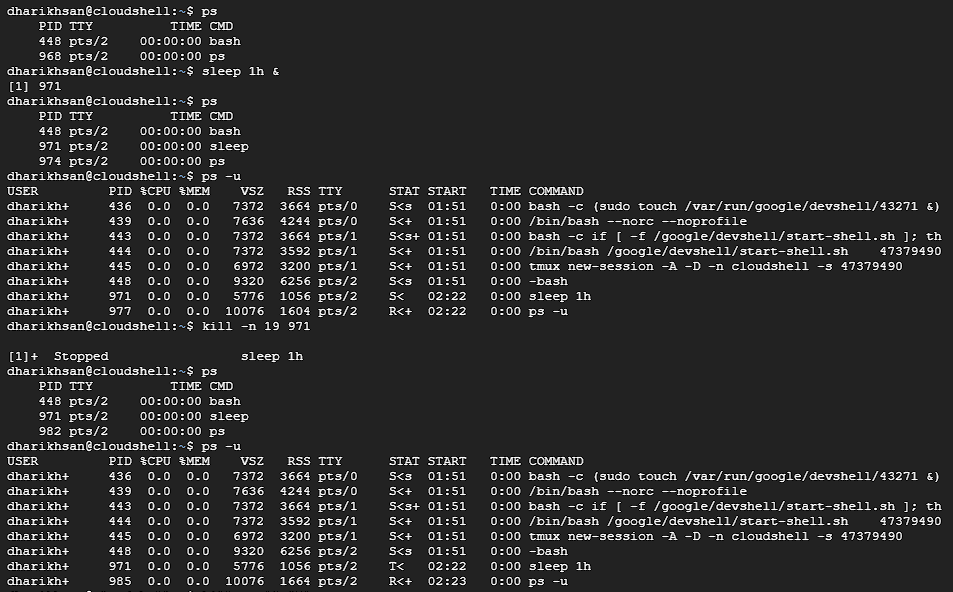
Nama: Daffa Harikhsan

NIM: 23/513044/PA/21918

**TUGAS 4**

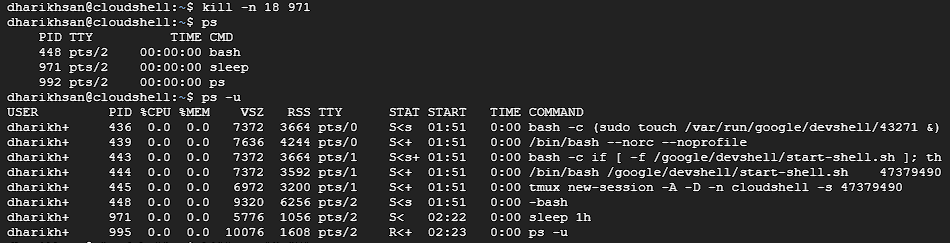
Activity 4.1

1). -n (menggunakan number), 19 (merujuk pada SIGSTOP)



Saat Perintah sleep 1h & akan menghentikan perintah tetapi di belakang layar, artinya sleep 1h akan tetap jalan meskipun kita melanjutkan perintah lagi dikarenakan ada tanda (&). Saat kita menggunakan perintah ps -u akan munjul di STAT bahwasannya proses PID 971 atau sleep 1h & itu S<, saat kita menggunakan perintah kill -n 19 971 akan muncul peringatan stop pada PID 971 atau sleep 1h & dan saat kita cek menggunakan command ps -u akan terlihat bahwasannya STAT PID 971 atau sleep 1h & itu berubah dari yang asalnya S< menjadi <T. S adalah sebuah status untuk sebuah proses yang sedang berjalan (Sleep) sedangkan T sendiri adalah sebuah status untuk sebuah kondisi Stop atau berhenti yang artinya proses SIGSTOP berhasil dilakukan yang membuat proses sleep 1h & berhenti.

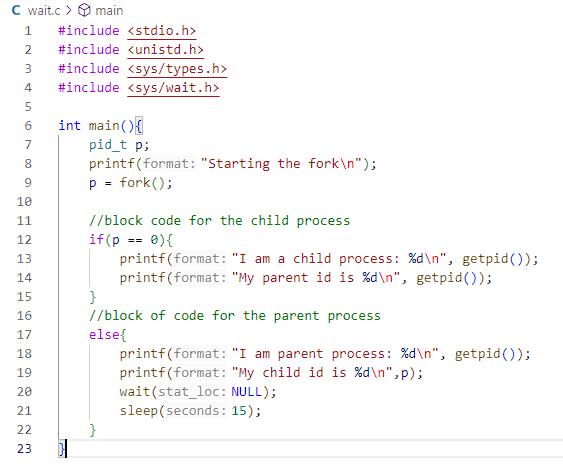
2). -n (menggunakan number), 18 (merujuk pada SIGCONT)

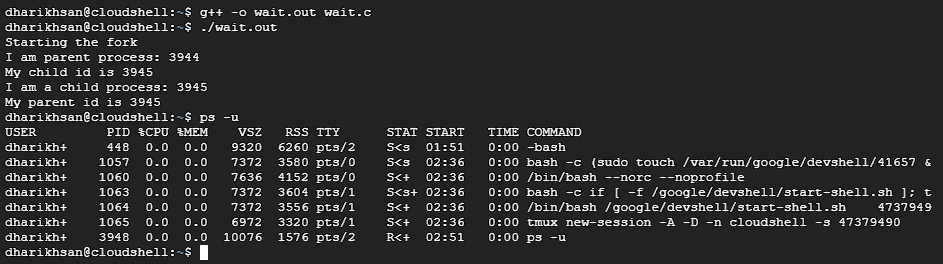


Saat Perintah kill -n 18 971 dijalankan pada terminal maka proses kita switch kembali menjadi aktif (T< menjadi S<). Maka pada saat itu command yang asalnya kita jalankan yakni sleep 1h & kembali aktif.

Activity 4.2

Before

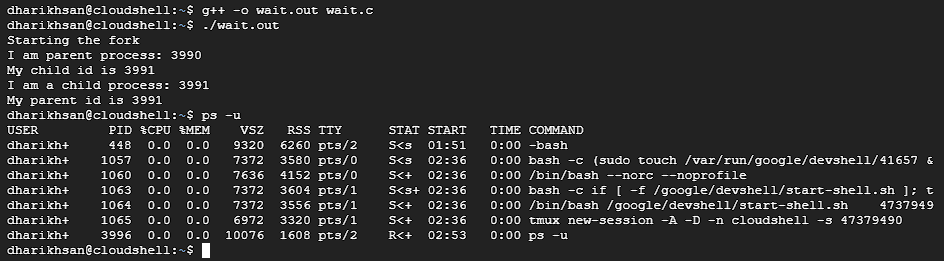




Sebelum kita modifikasi (menukar posisi wait setelah sleep) maka akan terlihat program tersebut pada parent process akan berhenti dan menunggu hingga child process selesai menjalankan semua instruksinya. Jika child process telah selesai, parent process akan lanjut ke baris berikutnya. Setelah child process selesai dan parent process melanjutkan, parent process akan tidur selama 15 detik sebelum proses berakhir. Efeknya Parent menunggu child process selesai dulu baru kemudian sleep selama 15 detik dan juga waktu eksekusi child process dan parent process terlihat lebih bersamaan, tetapi parent baru benar-benar selesai setelah 15 detik ditambah waktu eksekusi child process.

After modify





Setelah kita modifikasi menjadi sleep dulu setelah itu wait maka akan terlihat program tersebut pada parent process langsung tidur selama 15 detik setelah memulai child process tanpa menunggu apakah child process sudah selesai atau belum. Setelah 15 detik, parent process baru menunggu child process selesai jika child process belum selesai pada saat itu. Efeknya Parent process tidak menunggu child process secara langsung, melainkan tidur dulu selama 15 detik. Setelah tidur, parent process akan menunggu child process (jika masih berjalan). dan Jika child process selesai lebih awal (sebelum 15 detik), parent akan tetap tidur selama 15 detik sebelum mengeksekusi wait(NULL);. Hal ini membuat eksekusi child process dan parent process tampak tidak bersamaan, karena parent baru menunggu child process setelah sleep(15) selesai.