**EVALUASI**

1. Berdasarkan program **simpleFork.c**, jelaskan bagaimana proses mengetahui dirinya adalah induk atau anak?

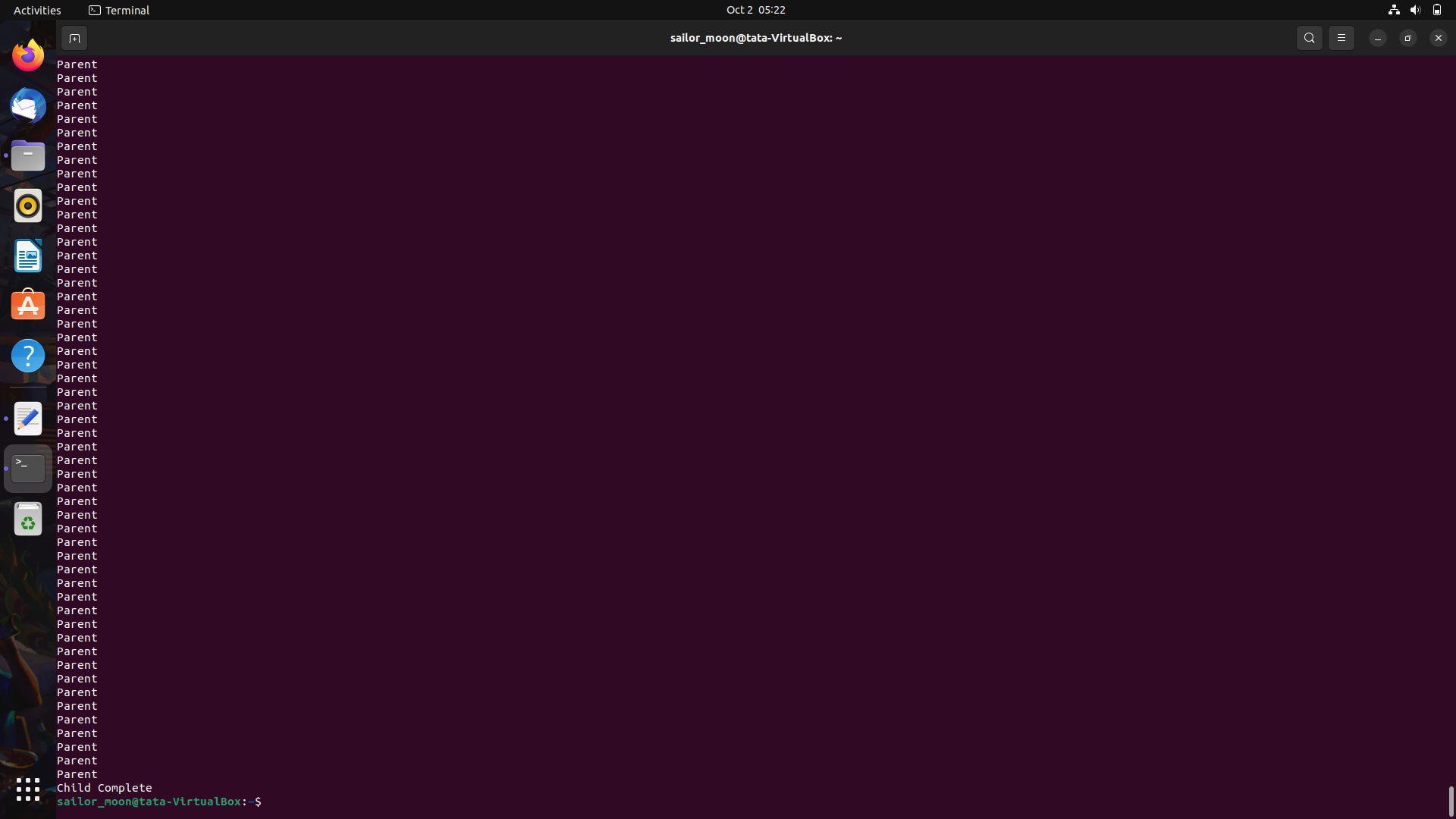
Jawab :

Dalam pemanggilan fungsi fork, akan dibuat proses baru yang disebut proses anak *(child process)*. Fungsi ini akan memberikan dua *return*, yaitu :

1. Proses anak mendapat *return* nilai 0
2. Proses induk mendapatkan *return* ID proses anak yang baru

Kemudian dicek nilai PID untuk mengetahui suatu proses termasuk induk atau anak, yaitu apabila PID bernilai kurang dari 0 maka proses fork gagal, apabila PID bernilai sama dengan 0 maka terjadi proses anak, sedangkan apabila PID bernilai lebih dari 0 maka terjadi proses induk.

Hasil :

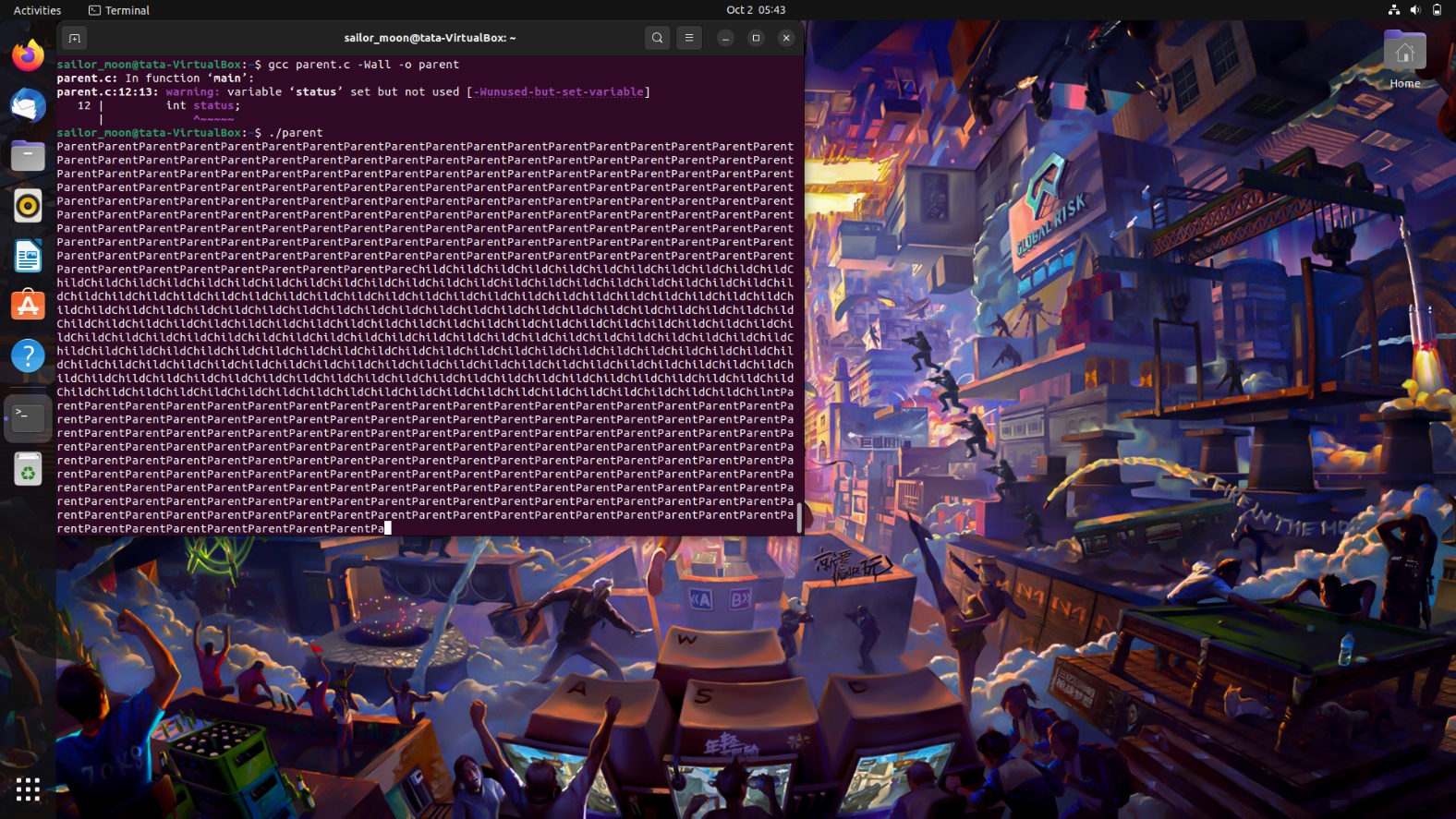


1. Berdasarkan eksekusi **parent.c**, jelaskan tujuan pemanggilan fungsi exec!

Jawab :

Fungsi exec berguna untuk menggantikan program saat ini dengan program baru dimana fungsi ini akan men-*spawn* proses anak baru. Berdasarkan eksekusi parent.c digunakan exec family, yaitu execl yang digunakan untuk menjalankan shell sebagai pelaksana proses anak.

Hasil :



1. Berikan kesimpulan cara penggunaan masing-masing keluarga exec berikut : execl, execlp. execv, execvp!

Jawab :

1. execl

Program yang dimuat secara full path dimana **path-command nya harus dimasukkan**. Execl merupakan non-V family sehingga tidak mem-*passing* argv, namun mem-*passing* semua argumen yang ada (*passing* argv[0], argv[1], argv[2]).

1. execlp

Program yang dimuat – **Path**, dimana parameter pertamanya berupa file **(harus sudah dipastikan berada pada path yang benar)**. Sama seperti execl, execlp merupakan non-V family sehingga semua argumen (argv[0], argv[1], argv[2]) yang ada harus di-*passing*.

1. execv

Program yang dimuat **hanya** full path. Execv merupakan V family maka yang di-*passing* adalah pointer menuju array & string argumennya.

1. execvp

Sama seperti execlp, execvp merupakan program yang dimuat dimana **path-nya dicari** (mencari file executable pada tiap path). Execvp merupakan V&P family maka dilakukan ***passing*** dengan membuat line menjadi array of string dengan pointer variabel argv. Variabel argv di-*passing* ke execute kemudian fungsi execute akan memanggil execvp.

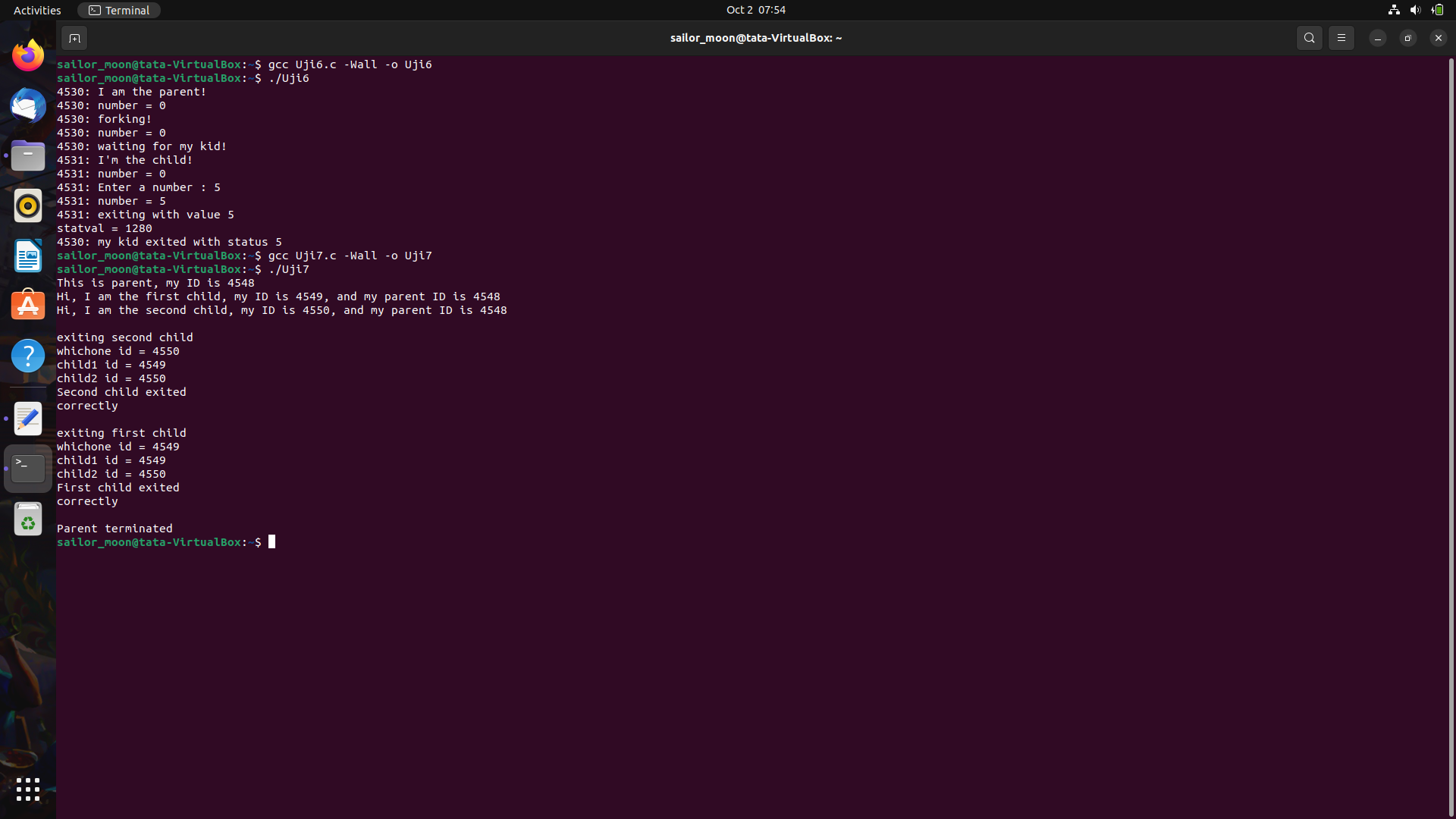
1. Apa fungsi argumen NULL pada argumen exec?

Jawab :

Pada exec family, fungsi yang memiliki huruf V menerima argumen untuk program yang dieksekusi sebagai NULL terminated array pointer ke string. Elemen terakhir dari array ini harus NULL dimana NULL digunakan untuk terminasi argumen-argumen dalam list.

1. Amati **uji6.c** apakah PID proses berubah ketika exec dieksekusi?

Jawab :



PID proses tidak berubah ketika exec dieksekusi karena exec family mengganti kerja suatu proses menjadi proses lain dengan mempertahankan sedikit data aslinya, yaitu PID dan PPID.

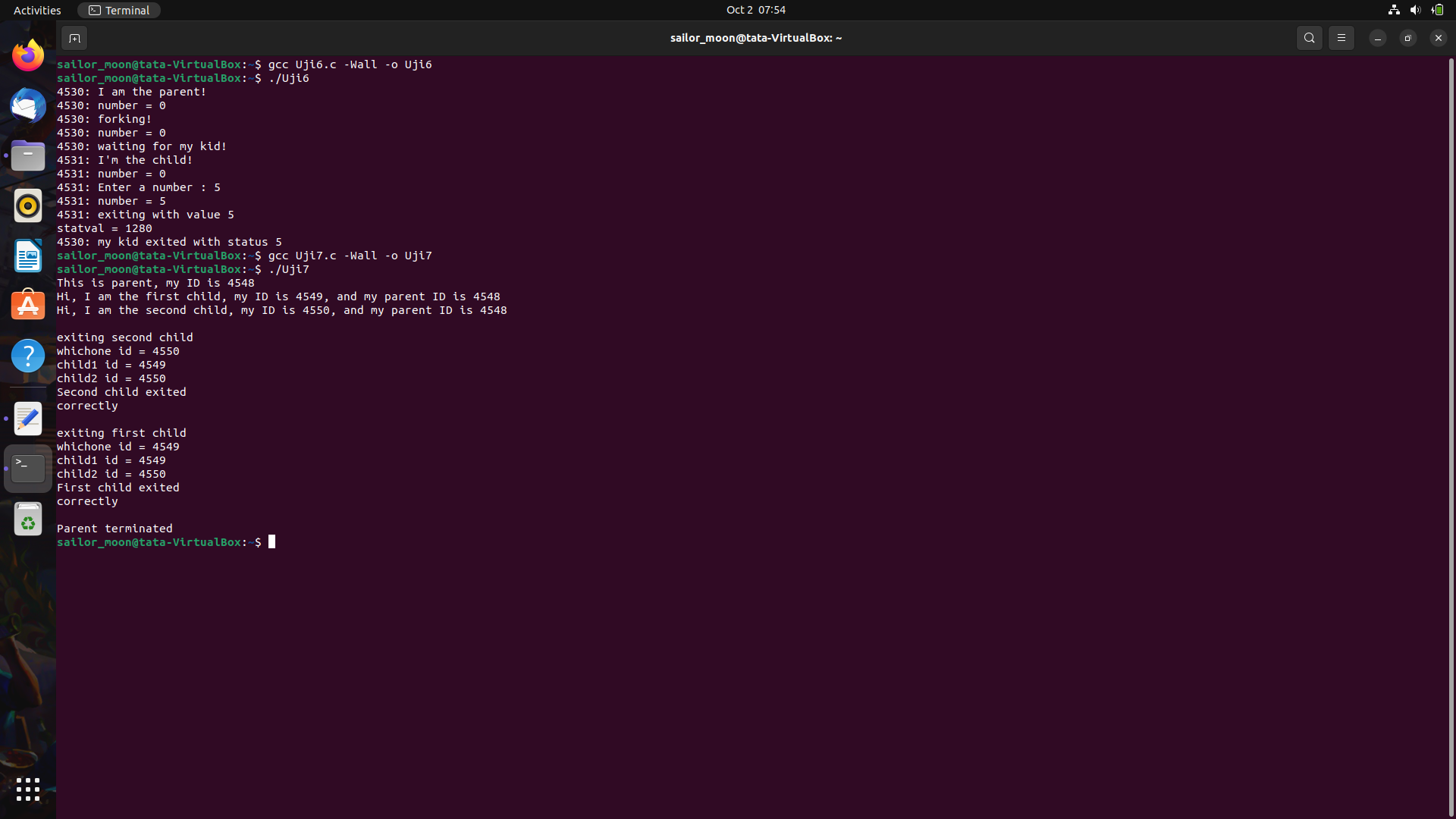
1. Berdasarkan **uji6.c** apakah terbukti bahwa sinyal yang dikirim oleh exit() ditangkap oleh wait()? Berikan penjelasannya!

Jawab :

Terbukti pada **uji6.c** bahwa sinyal yang dikirim oleh exit() ditangkap oleh wait(). Variabel statval akan menangkap sinyal yang dikirim oleh child menggunakan wait(&statval), proses child akan terminasi dan WEXITSTATUS(stataval) akan dipanggil kemudian meng-extract exit status sehingga muncul exit status di terminal yang nilainya sama dengan apa yang user inputkan. Hal ini dapat dilihat pada baris terakhir dimana proses parent berhasil mencetak nilai kembalian dari proses child dengan benar.

1. Berdasarkan **uji7.c** jelaskan siapa parent kedua child yang dibuat? Jelaskan pula bagaimana parent dapat mengenali child nya?

Jawab :



Parent dari kedua child (child1 dan child2), yaitu 4548. Melalui whichone=(int)wait(&status), parent dapat mengenali child-nya dengan cara mencocokkan whichone id dengan id dari masing-masing child.