

Forking Proses Pada Sistem Operasi Linux

Agustina Budiarti ²⁾

Abstrak

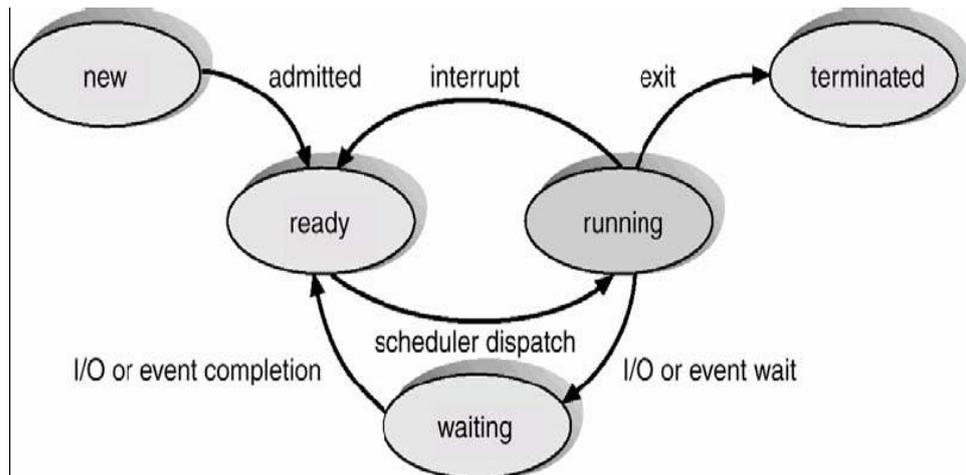
Dalam linux ataupun Unix dapat dilakukan pendjadwalan proses dimana proses dapat berjalan secara bersama – sama atau forking sehingga dapat melakukan banyak pekerjaan secara bersama – sama. Ibarat tangan, dalam proses diatas terdapat banyak tangan yang dapat menangani banyak pekerjaan sekaligus. Kalau diimplementasikan dalam proses di processor computer maka di dalam processor terjadi banyak forking untuk melakukan multi tasking terhadap proses yang diinputkan oleh user.

I. Pendahuluan

Sistem operasi mengeksekusi berbagai jenis program. Pada sistem batch program tersebut biasanya disebut dengan *job*, sedangkan pada sistem time sharing, program disebut dengan program user atau *task*. Beberapa buku teks menggunakan istilah *job* atau *proses*. Proses adalah program yang sedang dieksekusi. Eksekusi proses dilakukan secara berurutan. Dalam suatu proses terdapat *program counter*, *stack* dan daerah data.

Meskipun tiap-tiap proses terdiri dari suatu kesatuan yang terpisah namun adakalanya proses-proses tersebut butuh untuk saling berinteraksi. Satu proses bisa dibangkitkan dari output proses lainnya sebagai input. Pada saat proses dieksekusi, akan terjadi perubahan status. Status proses didefinisikan sebagai bagian dari aktivitas proses yang sedang berlangsung saat itu.

²⁾ Staf Pengajar STMIK Sinar Nusantara Surakarta



Gambar 1. Diagram status proses.

Status proses terdiri dari :

- a. *New*: proses sedang dibuat.
- b. *Running*: proses sedang dieksekusi.
- c. *Waiting*: proses sedang menunggu beberapa *event* yang akan terjadi (seperti menunggu untuk menyelesaikan I/O atau menerima sinyal).
- d. *Ready*: proses menunggu jatah waktu dari CPU untuk diproses.
- e. *Terminated*: proses telah selesai dieksekusi.

II. Perumusan Masalah

Dalam Penelitian ini masalah yang dibahas hanya dibatasi untuk melihat proses pada Linux dengan Forking terhadap PID dan PPID.

III. Tujuan dan Mafaat

Untuk Mengetahui proses pada LINUX sehingga dapat diketahui alokasi memori pada linux.

IV. Metode Penelitian

Metode penelitian yang digunakan adalah dengan melakukan studi kepustakaan dan browsing di internet serta melakukan uji coba dengan melakukan coding dengan Bahasa Pemrograman C di Laboratorium.

V. Pembahasan

5.1. Running Program

```
Shell - Konsola
Session Edit View Bookmarks Settings Help
Shell

PARENT
Pid Parent = 8309
Parent -> Sekarang menunggu child selesai...

CHILD 2
Pid Child 2 = 8313
Parent ID dari child 2 adalah = 8309
Ini Proses Child 2 ....

    CHILD 1.1.1
    Pid Child 1.1.1 = 8312
    Parent ID dari child 1.1.1 adalah = 8311
    Ini Proses Child 1.1.1 ....

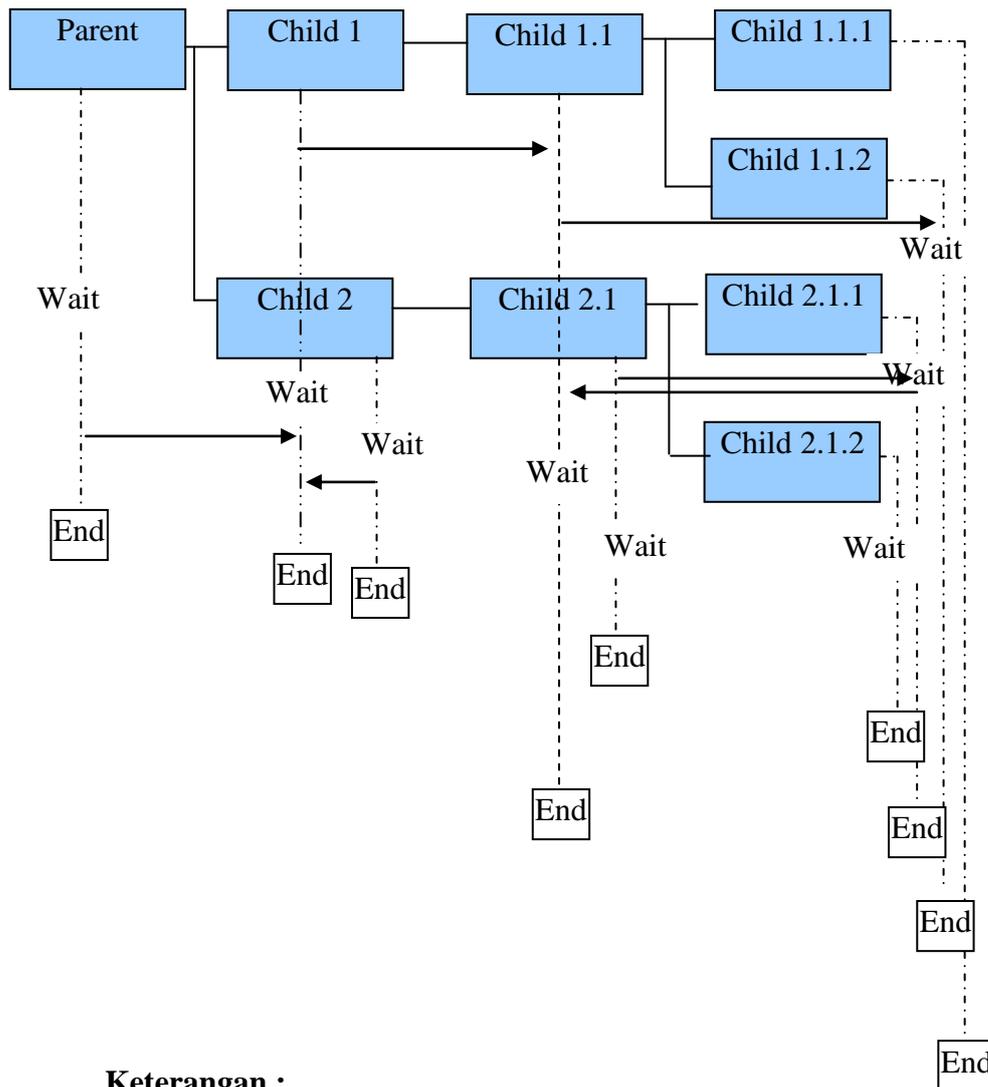
    CHILD 1.1.2
    Pid Child 1.1.2 = 8315
    Parent ID dari child 1.1.2 adalah = 8311
    Ini Proses Child 1.1.2 ....

    CHILD 2.1
    Pid Child 2.1 = 8314
    Parent ID dari child 2.1 adalah = 8313
    Ini Proses Child 2.1 ....

        CHILD 2.1.1
        Pid Child 2.1.1 = 8316
        Parent ID dari child 2.1.1 adalah = 8314
        Ini Proses Child 2.1.1 ....

        CHILD 2.1.2
        Pid Child 2.1.2 = 8317
        Parent ID dari child 2.1.2 adalah = 8314
        Ini Proses Child 2.1.2 ....
```


5.3. Cara Kerja Program



Keterangan :

1. Terjadi proses fork, untuk membentuk child1 kemudian parent menunggu (wait)
2. Child 1 melakukan fork untuk membentuk child 1.1, disini child1 berfungsi sebagai parent bagi child1.1, kemudian menunggu.
3. Child1.1 melakukan fork untuk membentuk child1.1.1, disini child1.1 berfungsi sebagai parent bagi child1.1.1 kemudian menunggu.

4. Selanjutnya child1.1 melakukan forking lagi untuk membentuk child1.1.2, kemudian menunggu
5. Selanjutnya parent melakukan forking lagi untuk membentuk child2 kemudian menunggu
6. Child2 melakukan fork untuk membentuk child2.1 maka disini child2 berfungsi sebagai parent bagi child2.1, kemudian menunggu
7. Child2.1 melakukan fork untuk membentuk child2.1.1, maka otomatis child2.1 berfungsi sebagai parent dari child2.1.1, kemudian menunggu
8. Child2.1 melakukan forking yang kedua kali untuk membentuk child 2.1.2, kemudian menunggu.
9. Setelah itu proses di child1.1.1 selesai
10. Proses di 1.1.2 selesai
11. Proses di 1.1 selesai
12. Proses di 2.1.2 selesai
13. Proses di 2.1.1 selesai
14. Proses di child1 selesai
15. Proses di child2 selesai
16. Proses Parent Selesai

5.4. Analisa Program

Terjadi proses fork, untuk membentuk child1 kemudian parent menunggu (wait) , Child 1 melakukan fork untuk membentuk child child 1.1, disini child1 berfungsi sebagai parent bagi child1.1, kemudian menunggu. Selanjutnya Child1.1 melakukan fork untuk membentuk child1.1.1, disini child1.1 berfungsi sebagai parent bagi child1.1.1 kemudian menunggu. Selanjutnya child1.1 melakukan forking lagi untuk membentuk child1.1.2, kemudian menunggu Selanjutnya parent melakukan forking lagi untuk membentuk child2 kemudian menunggu Child2 melakukan fork untuk membentuk child2.1 maka disini child2 berfungsi sebagai parent bagi child2.1, kemudian menunggu Child2.1 melakukan fork untuk membentuk child2.1.1, maka otomatis child2.1 berfungsi sebagai parent dari child2.1.1, kemudian menunggu Child2.1 melakukan forking yang kedua kali untuk membentuk child 2.1.2.

Sehingga akan terbentuk tree sebagai berikut :

```
Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
Shell
`-usb-storage(1619)
[tamu@segersumyah~]$
[tamu@segersumyah~]$ pstree -p
init(1)-+-crrond(5096)
|-cupsd(5072)
|-dcoerver(5848)
|-events/0(3)
|-hpiod(5050)---hpiod(5055)---hpiod(5056)
|-inetd(5107)
|-kaccess(5872)
|-kded(5853)---kded(5860)---kded(5861)
|-kdeinit(5845)-+-artsd(5867)---artsd(5900)
|   |-kio_file(5895)
|   |-kio_file(5958)
|   |-klauncher(5850)
|   |-konsole(7542)---bash(7547)---agus(8309)-+-agus(8310)---agus(8311)-+-agus(8312)
|   |                                     |                                     `--agus(8315)
|   |                                     `--agus(8313)---agus(8314)-+-agus(8316)
|   |                                                         `--agus(8317)
|   |-konsole(7737)---bash(7739)---pstree(8525)
|   |-kwin(5878)
|   |-kwrite(7382)---kwrite(7478)
|   `--soffice(6777)---soffice.bin(6806)---soffice.bin(6816)-+-soffice.bin(6818)
|                                                           |-soffice.bin(6820)
|                                                           |-soffice.bin(6824)
|                                                           `--soffice.bin(6831)
|-kdesktop(5883)
|-kdm(5220)-+-X(5725)
|   `--kdm(5726)---startkde(5788)---kwrapper(5873)
|-khelper(4)
|-kicker(5885)
|-kio_uiserver(6457)
|-kLipper(5887)
|-klogd(1547)
|-knotify(5894)
```



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell
CHILD 2.1.1
Pid Child 2.1.1 = 8316
Parent ID dari child 2.1.1 adalah = 8314
Ini Proses Child 2.1.1 ....

CHILD 2.1.2
Pid Child 2.1.2 = 8317
Parent ID dari child 2.1.2 adalah = 8314
Ini Proses Child 2.1.2 ....

Proses Child 1.1.1 selesai ....

Proses Child 1.1.2 selesai ....

Proses Child 1.1 selesai ....

Proses Child 2.1.1 selesai ....

Proses Child 2.1.2 selesai ....

Proses Child 2.1 selesai ....

Proses Child 1 selesai ....

Proses Child 2 selesai ....

Proses Parent selesai ....

[tamu@segersumyah~]$
```

VI. Kesimpulan

Dari proses diatas dapat disimpulkan bahwa dalam linux ataupun Unix dapat dilakukan pendjadwalan proses dimana proses dapat berjalan secara bersama – sama atau forking sehingga dapat melakukan banyak pekerjaan secara bersama – sama. Ibarat tangan, dalam proses diatas terdapat banyak tangan yang dapat menangani banyak pekerjaan sekaligus. Kalau diimplementasikan dalam proses di prosessor computer maka di dalam prosesor terjadi banyak forking untuk melakukan multi tasking terhadap proses yang diinputkan oleh user.

Daftar Pustaka

- *Pengantar Sistem Operasi Komputer*, Masyarakat Digital Gotong Royong(MDGR), 2003- 2005
- www.wikipedia.com