Rocky Arkan Adnan Ahmad

1806186566

SysProg-A

Hal yang sudah dipelajari dan dipahami:

1. Dsaemon
   Daemon sebelumnya sudah diajari di minggu sebelumnya namun di minggu ini diajari lebih dalam lagi mengenai daemon. Daemon adalah background process yang berjalan sejak system dinyalakan sampai dimatikan. Daemon mempunyai ciri-ciri: mempunyai init sebagasi parentnya, menutup semua file descriptor, tidak mempunyai controlling terminal, merupakan session dan group leader, dan dieksekusi ketika boot dan selalu hidup sampai system dimatikan. Karena daemon menutup semua file descriptor yang tidak mempunyai controlling terminal, maka ia tidak bisa menerima input dan mengirim output ke terminal. Biasanya, daemon mengirim log ke dalam system logger.

2. Foreground dan background process
   Secara default ketika menjalankan sebuah command di shell maka process untuk command tersebut adalah foreground process. Foreground process adalah process yang sedang dijalankan dan berinteraksi dengan user saat ini, dimana user menunggunya sampai selesai. Lalu, Menjalankan command dengan & akan membuat process untuk command tersebut menjadi background process. Background process adalah process yang berjalan tapi tidak berinteraksi dengan user tanpa mengeblock shell atau shell tidak perlu menunggu sampai process tersebut selesai berjalan. Hal ini dilakukan dengan cara menutup stdin dari process, sehingga sekarang process tidak bisa menerima output dari terminal. Lalu, ketika menjadi background process maka process tersebut akan masuk ke dalam daftar jobs dari shell yang menjalankannya. Ketika menjadi jobs, maka process akan diberi signal SIGHUP ketika controlling terminal dimatikan. Selain itu kita bisa mengecek status dari process tersebut (apakah running, finished, atau terminated) dengan command jobs. Ketika process selesai dijalankan, maka shell akan memberi tahu kalau process selesai dijalankan.

3. Nohup dan disown
   Command disown adalah command yang mengubah sebuah background process yang merupakan jobs menjadi bukan merupakan jobs dari shell tersebut. Ketika background process bukan merupakan jobs, maka terminal tidak akan mengirim signal SIGHUP ke process tersebut ketika terminal dimatikan sehingga process tidak ikut mati ketika terminal dimatikan, dan juga kita tidak bisa mengecek status process dengan command jobs dan ketika process selesai dijalankan shell tidak memberi tahu kalau sudah selesai. Tapi walaupun menjadi bukan jobs, controlling terminal dari process tersebut masih merupakan terminal yang menjalankannya dan stdout dan stderr dari process masih mengarah ke terminal sehingga ketika mengeprint output masih akan mengirim ke controlling terminalnya. Lalu, walaupun controlling terminalnya tidak mengirim signal SIGHUP, kita masih bisa mengirim signal SIGHUP secara manual kepada process tersebut untuk mematikannya.

Lalu, command nohup akan membuat process menjadi immune dari signal SIGHUP, sehingga kita tidak bisa mematikannya dengan signal SIGHUP dan untuk mematikannya harus dengan signal lain (SIGKILL atau SIGTERM misalnya). Lalu process juga menutup stdout dan stderr, sehingga sekarang output dari process tersebut diredirect ke sebuah file yaitu nohup.out. Perbedaan dari kedua command adalah nohup sendiri tidak membuatnya menjadi background process, sehingga harus dijalankan dengan command & untuk menjalankannya sebagai background process. Lalu, ketika menjadi background process maka command nohup masih menjadi jobs dari shell sehingga status dari process masih bisa dimonitor dengan command jobs dan juga ketika process selesai maka shell akan memberi tahu kalau process sudah selesai, dan juga shell akan mengirim signal SIGHUP ketika dimatikan. Walaupun begitu, karena process immune terhadap SIGHUP, maka process tetap berjalan walaupun diberi signal SIGHUP oleh controlling terminal.

4. Process group dan session
   Process group adalah koleksi dari satu atau lebih process. Semua process termasuk ke dalam suatu process group, dan suatu process group diwakili oleh process group leader yaitu process yang memiliki PID yang sama dengan process group tersebut (misal process group memiliki PGID 100, maka process dengan PID 100 adalah process group leadernya). Lalu, untuk suatu process group hanya ada 1 foreground process dan process lain yang bukan merupakan foreground process adalah background process.

   Lalu, ada juga Session. Session adalah koleksi dari satu atau lebih process group. Seperti process group, ketika suatu process pertama kali dibuat maka akan menggunakan session dari parentnya, semua process pasti mempunyai sebuah session, dan session leader adalah process yang pertama kali dibuat untuk session tersebut yang juga merupakan nama dari session tersebut. Lalu, semua session mempunyai sebuah controlling terminal, yang bisa dibilang controlling terminal tersebut merupakan controlling terminal untuk semua process & process group dari session tersebut.

   Ketika memanggil setsid(), maka akan dibuat sebuah session baru dengan process yang memanggilnya menjadi session leader baru. Lalu, karena membuat session baru, maka dibuat juga process group baru untuk session tersebut yang mempunyai leader process yang memanggilnya. Ketika session baru dibuat maka process tersebut akan tidak mempunyai controlling terminal karena controlling terminal hanya mengontrol untuk 1 session.

Hal yang dipelajari namun belum dipahami:

Untuk minggu ini belum ada materi yang belum saya pahami.