****

**Tugas Tulisan Ilmiah Proses**

**Praktikum Sistem Operasi**

**PTI Kelas A**

**Nama Mahasiswa:**

Mohammad Dwi Anshor Al Ghifari- 235150607111008

**Asisten Praktikum:**

Adityo Nugroho 225150601111004

Keira Kayla Aqila 225150607111021

**Dosen:**

Aswin Suharsono, S.T., M.T.



**Program Studi Pendidikan Teknologi Informasi**

**Fakultas Ilmu Komputer**

**Universitas Brawijaya**

**202**

1. Kesimpulan

Buatlah dalam bentuk paragraph dalam menjawab pertanyaan ini

1. Jelaskan apa itu proses!
2. Jelaskan secara rinci dan singkat perbedaan system call exec() dan fork()!
3. Bagaimana urutan proses saat menghidupkan computer?
4. Referensi

Buatlah referensi dengan format sesuai dengan referensi pada penulisan skripsi <<http://file-filkom.ub.ac.id/fileupload/assets/upload/filemanager/Skripsi/PanduanSkripsiFilkom-v3.0.pdf>>

**PROSES PADA SISTEM OPERASI LINUX**

1. **PROSES UTAMA YANG DITENTUKAN TELAH BERJALAN SESAAT SETELAH SISTEM OPERASI LINUX BERHASIL DIJALNKAN**

Proses utama yang ditemukan telah berjalan sesaat setelah system operasi Linux berhasil dijalnkan adalah proses init. Init adalah root/ parent dari semua proses yang dijalnkan pada Linux. Proses yang perama untuk memulai init adalah skrip /etc/rc.d/rc/sysinit. Run-level yang ditentukan, srip dieksekusi untuk memulai proses tertentu agat dapat menjalankan system dan membuatvsistem lebih fungsional dan proses inijuga proses pertama yang dimulai oleh kernel linux setelah proses booting. Tugas dari init adalah untuk mengawasi dan mengelola proses yang dibutuhkan dan juga bertugas untuk menjalankan sebuah system operasi dengan penuh.

Proses init dapat diimplementasikan dengan berbagai varian yaitu seperti: SysV init, Upstart, atau system semua itu juga tergantung pada distribusi :inux yang digunakan . Tetapi, fungsi init tetap sama pada semua varian yaitu dengan memulai dan mengawasi proses lainnya. Pross init memilikitaggung jawab yaitu mengani proses yang telah selesai tetapi masih memilki entri dalam proses kernel.

1. **RELASI ANTAR PROSES DAN KETRKAITANNYA DENGAN SISTEM CALL FORK() DAN EXEC()**

Proses dalam sebuah system oprrasi Linux bias mewakili hubugan yang sangat beragam satu sama lain, dan dapat berjalan secara snduru atau saling bergantung dalam menjalankannya . Proses dalam operasi system Linux saling berhubungan. Ketika sebuah proses induk memanggil system fork() dengan mencptakan salinannya. Proses induk merupakan sebuah parent process dan child proses. Child proses menggunakan system call exec() untuk mengganti kode program dengan yang baru sehingga dapat berhubungan dengan proses induknya.

System call merupakan sebuah layaan langsung system operasi ke pemrograman atau sebuah pemanggilan aplikasi untuk mendapatkan sebuah layanan yang disediakan oleh System Operasi. System call merupakan sebuah runtime yang digunakan untuk keperluan yang spesifik, system inidiperlukan untuk memajemen sebuah prose yang sedang berjalan. Melalui dari sebuah system call program dapat dumunta layanan tertentu seperti akses ke perangkat keras, pengelolan memori, pembuatan proses baru, dan operasi I/O

* System Call fork() memiliki sebuah fungsi untuk membuat proses yang baru.System call fork() ini dibagi menjadi dua yaitu proses induk dan proses turunan. Menciptakan child proses yang memiliki identitas unik dan brbagi kode program, serta memiliki alamat memoi yang terpisah. System ini berguna dalam sebuah pengembangan aplikasi yang memerlukan proses paralel. Dapat menjalankan tugas secara bersama engan proses induk.
* System Call exec() memiliki sebahb fungsi untuk membuat proses turunan yang telah terbentuk dan memiliki instruksi yang berbeda dengan proses induknya. Menjalankan program yang berbeda tanpa kehilangan identitasnya System ini berguna ketika sebuah proses ingin menjlankan program yang berbeda tanpa kehilangan identitasnya. Dapat menjalankn program baru tanpa membuat proses yang baru.

Hubungan relasi dari system call fork() dan system call exec () yaitu child akan dibentuk oleh parent menggunakan system call fork(). Setelah fork() dipanggil maka parent akan kembali berjalan secara parallel dengan child, Child dapat memanggil fork() digunakan untuk memanggil child lainnya.

System call exec() digunakan untuk setelah fork() untuk mengganti alamat memory dari proses program yang baru. System call dapat memanggil binary file tertentu yang berada dimemory fan mengaksesnya sebagai child. Selanjutnya, parent akan running kembali setelah child selesai dijalankan, maka parent dan child akan berjalan secara parallel.

Kesimpulan

1. **PROSES**

Dalam ilmu komputer, peoses merupakan suatu keadaan sebuah program kompuer yang dieksekusi atau dijalankan. Proses juga merupakan unit terkecil yang secra individu memiliki sumber daya yang dijadwalkan oleh system operai tersebut. Sebuah program dapat sebuah dimuat dari media penyimpanan dan instance dari program yang dijalankannya. Proses memilliki memorinya sendiri, yang disebut dengan alamat proses. Ruang alamat proses mempunyaiarea penting yaitu are a teks dan area data. Beberapa proses yang terdapat pada system operasi linux yaitu:

* Interactive : merupakan sebuah proses yang dimulai dan diatur oleh shell. Nisa berjalan di luar atau di berjalan di background.
* Daemon: sebuah proses yang dapa dimulai ketika Linux dalam proses booting. Proses ini berjalan secara background, dan dapat menganguperminaan dari proses lainnya jika tidak ada request.
* Batch: Proses ini merupakan proses yang tidak memiliki hubungan dengan terminal dan proses ini menunggu untuk dieksekusi.

Pada komputer terdapat banyak proses yang berjalan sehingga proses tersebut perlu dikelola agar tidak terjadi masalah pada pross tersebut . Sebuah proses pasti membutuhkan sumber daya yang dapat menyelesaikan tugasnya, sumber daya tersebut dapat berupaCPU Time. File berkas dan juga perangka I/O memiliki kebutuhannya masing-masing pada proses Sistem Operasi juga bertugas untuk memanajemen proses tersebut. Beberapa tanggung jawab system operas berikut :

* Membuat dan menghapus proses
* Meyediakan mekanisme proses sinkronisasi
* Menyediakan mekanisme proses komunikasi
* Menydiakan mekanisme agar tidak terjadi feedback pada saat proses berjalan
* Melanjutkan dan memindahkan proses

Ketika sebuah proses dieksekusi, maka akan terjadi sebuah perubahan status. Yang dapat didefinisikan bagian dari aktidfitas yang sedang berlangsung pada saat itu. Beberapa jenis status yang dijalankan:

* Status New: proses sedang dibuat
* Status Ready: proses siap dieksekusi tetapi CPU belum tersedia. Dimana CPU tersebut sedang menjalankan proses lain.
* Status Writing: Proses sedang menunggu suatu event tertentu

Sebuah proses memiliki satu atau lebih thread. Jika pada suatu proses memiliki satu thread maka proses tersebut akan memilikiaturan yang sama seperti lainnya yaitu mengijinkan single task pada waktu tertentu, tetapi jika sebuah thread memiliki lebih dari satu makn task tersebut akan berjalan secara langsung atau bersamaan.

1. **PERBEDAAN SYSTEM CALL FORK() DAN SYSTEM CALL EXEC()**

System call exec() dan system call fork() merupkan dua system call yang digunakan untuk membuat proses baru dalam system operasi.

1. System Call Exec()

System call exec () digunakan dalam sebuah system operai untuk mengganti proses yang berjalan dengan proses baru yang memiliki instruksi yang berbeda.

* Tujuan utama

Menggantikan program lama dengan program baru

* Aksi Yang Dilakukan

Mengganti proses yang berjalan menjadi program yang baru

* Kembalian Nilai

Tidak mengembalikan nilai ke proses pemanggil

* Fungsi Utama

Untuk menjalankan program yang baru untuk menggantikan program yang berjalan di sebuah proses tersebut. Serta digunakan pada suatu program ingin menjalankan program yang berbeda tetapi menggunakan proses yang sama.

* Pergantian Proses

Proses yang berjalan akan dihapus dan diganti oleh program baru yang dijalankan. Ruang alamat memory, koe program, dan sumber daya dari proses lama diganti dengan program yang baru.

* Kembalian Nilai

Digunakan untuk mengambalikan nilai ke proses pemanggil. Bila terjadi kesalahan pada program baru maka system akan mengembalkan kode kesalahan dan proses akan mengalami kegagalan.

1. System Cal Fork()

Merupakan sebuah system fundamental dalam system operasi sehingga digunakan untuk membuat proses baru dengan menduplikasi peoses berjalan.

* Tujuan

Bertujuan untuk membuat proses baru dari proses yang sedang berjalan

* Aksi yang Dilakukan
* Kembalian Nilai
* Fungsi Utama
* Duplikasi Proses
* Kembalian Nilai

Referensi

* Silberschatz, A., Galvin, P.B., & Gagne, G. (2018). Operating System Concepts, Tenth Edition. John Wiley & Sons, Inc.
* Love, R. (2013). Linux Kernel Development. Addison-Wesley Professional.
* Kerrisk, M. (2010). "The Linux Programming Interface: A Linux and UNIX System Programming Handbook." No Starch Press.
* Bovet, D. P., & Cesati, M. (2005). "Understanding the Linux Kernel, Third Edition." O'Reilly Media, Inc.
* Beck, M. (2011). "Linux Kernel Internals." Addison-Wesley Professional.
* Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th Edition). Pearson.
* Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th Edition). Pearson.