PR 1

Muhammad Irfan Amrullah / 1706039585

Sysprog

1. Iya, direktori itu berdasarkan Filesystem Hierarchy Standard(FHS). Semua file dan directory muncul dibawah directory root /, bahkan ketika dia ditaruh di device virtual atau fisik yang berbeda.
2. Absolut path: Sebuah absolute path di artikan sebagai lokasi spesifik sebuah file atau direktori dari direktori root(/).

Alternative path: Sebuah relative path di artikan sebagai path yang bergantung pada present working directly(pwd). Di mulai dari directory kamu sekarang dan tidak pernah dimulai oleh / .

1. Ada, tool tersebut adalah file. Perintah paling sederhananya ada “file (nama file yang ingin diketahui jenisnya)” – tanpa tanda kutip. Semua fitur dapat dibaca pada man file.
2. Dapat. /home/joko/surat

Kelompok Soal Kedua

1. Informasi yang didapat Nida adalah jumlah major dan minor dari device-device yang di partisi. Major number menentukan device driver. Minor number adalah nomor identifikasi unik dari sebuah device.
2. /proc/diskstats, memberikan informasi (termasuk nomer device) untuk tiap logical disk devices.

/proc/modules, berisi list dari modul kernel yang sedang di load. Dia juga memberikan indikasi dari dependensi.

/proc/partitions, berisi list nomer device, ukuran, dan /dev yang dikenali kernel sebagai partisi yang sudah ada

Menurut saya hal paling penting adalah /proc/modules, kita dapat melihat modul mana yang sedang di load, apakah modul itu bergantung pada modul lain, dan dapat melakukan debugging dari sini menggunakan profiling tools seperti oprofile.

Kelompok Soal Ketiga: System Calls

1. - A system calls pertama mentrigger fungsi wrapper C library yang mana tugasnya melakukan menaruh informasi yang dibutuhkan untuk ditaruh di CPU registers. Nomer System Call ditaruh di register %eax.

* Setelah itu, fungsi wrapper yang sama mentrigger sebuah intruksi trap ‘int 0x80’. Instruksi ini menukar mode processor dari mode user ke mode kernel dan mengeksekusi kode yang disimpan di 0x80.
* Habis itu karena sekarang processor ada di mode kernel, kernel memanggil sebuah fungsi handler system\_call() untuk mengatasi trap ini.
* Fungsi handler ini mengcopy semua informasi relevan yang disimpan di register CPU ke stack kernel, memvalidasi nomer system call, menggunakan nomer system call untuk mencari system call service routine yang berkaitan dengan argument yang sudah diberikan dari user space
* System call service routine yang sudah dipanggil trap handler akan mengecek apakah argument yang diberikan valid baru kemudian dia mengerjakan perinta yang sudah diminta oleh aplikasi.
* Ketika system call service routine sudah memprosesnya, control dikembalikan ke fungsi trap handler yang mana akan mengembalikan fungsi wrapper ke mode user
* Berdasarkan nilai return dari trap handler, fungsi wrapper akan mengeset variable ‘errno’ dan mengembalikan -1 ketika ada error.
* Nilai ‘errno’ dihasilkan dari menegasikan nilai return yang diberikan trap handler.
* Aplikasi kemudian dapat menggunakan strerror() API dengan ‘errno’ untuk mendapatkan informasi error dalam bentuk string yang dapat dibaca manusia.

1. – System call pertama adalah gettimeofday(). Berfungsi untuk mengambil waktu dan time zone.

Outputnya seperti berikut: ./time 1000 26

* System call kedua adalah getrusage(). Berfungsi untuk ngembaliin resource usage measures.

Contoh output:

CPU TimeL 0.000000 sec user, 0.000526 sec system