

Teknologi Virtualisasi

#Praktikum Fuse Filesystem

Henry Saptono, M.Kom
**STT Terpadu Nurul
Fikri**

Filesystem In Userspace (FUSE)

- Filesystem dalam ruang pengguna.
- Kerangka kerja independen OS (*abstraction layer*) untuk mengimplementasikan filesystem
 - Komponen Kernel : Antarmuka Filesystem OS sebenarnya
 - Komponen Userland : FUSE API / library
 - Filesystem modules: portable
- Semua API pengguna tersedia untuk kode fs Anda:
 - Menulis filesystem dengan Python, Perl, Java, ..
 - Debug filesystem menggunakan userspace tools
- Tersedia untuk Linux, * BSD, Solaris, MacOSX

Filesystem In Userspace (FUSE)

- FUSE memungkinkan Anda mengembangkan filesystem yang berfungsi penuh yang memiliki pustaka API sederhana, dapat diakses oleh pengguna yang tidak memiliki hak istimewa, dan memberikan implementasi yang aman. Dan, untuk melengkapi semua ini, FUSE memiliki rekam jejak stabilitas yang telah terbukti.
- Dengan FUSE, Anda dapat mengembangkan filesystem sebagai binary executable yang terhubung ke pustaka FUSE - dengan kata lain, kerangka filesystem ini tidak mengharuskan Anda mempelajari filesystem internal atau program modul kernel.

Membuat Filesystem FUSE (dengan python)

- Untuk membuat filesystem di FUSE, Anda perlu menginstal modul kernel FUSE dan kemudian menggunakan library FUSE dan API untuk membuat filesystem Anda.
- Install library FUSE:
 - **# sudo apt-get install libfuse-dev**
- Install API fuse untuk python
 - **# sudo pip install fusepy**
- Download dan modify sample kode python untuk membuat filesystem fuse , di **<https://github.com/skorokithakis/python-fuse-sample/blob/master/passthrough.py>**

Contoh Modifikasi

- Dari sample yang diperoleh dari <https://github.com/skorokithakis/python-fuse-sample/blob/master/passthrough.py>
- Berikut ini contoh modifikasi di dalam pemanggilan fungsi **unlink()**. Fungsi unlink akan dipanggil ketika terjadi proses penghapusan file. Jadi jika Anda ingin melakukan suatu operasi saat terjadi penghapusan file didalam kode filesystem fuse Anda, maka lakukan modifikasi di fungsi unlink tersebut.

Contoh Modifikasi

- Kode awal/asli dari sample:
 - **def unlink(self, path):**
 - **return os.unlink(self._full_path(path))**
- Kode yang dimodifikasi:
 - **def unlink(self, path):**
 - **import shutil**
 - **shutil.copy2(self._full_path(path), "/tmp/trash**
%s"%path)
 - **return os.unlink(self._full_path(path))**

Menjalankan kode program

- Setelah Anda memodifikasi kode program `passthrough.py` jalankan/eksekusi dengan perintah:

- `$ mkdir /tmp/trash`

- `$python passthrough.py /direktori /mount-point`

#catatan: `direktori /direktori` diasumsikan adalah direktori yang ada di komputer Anda. Dan `direktori /mount-point` adalah direktori target mounting

- Anda akan melihat semua file di bawah/dalam direktori `/mount-point` dan dapat memanipulasinya persis seperti di sistem file aslinya.

Pertanyaan

- Jelaskan apa yang terjadi jika pada direktori `/mount-point` terdapat sebuah file kemudian file tersebut Anda hapus. !