

FUSE - systemy plików w przestrzeni użytkownika

Paweł 'pbm' Szubert

5 marca 2009

FUSE - co to takiego?

FUSE

Filesystem in UserSpace - umożliwia tworzenie systemów plików działających w przestrzeni użytkownika, zamiast w przestrzeni jądra.

Tradycyjne systemy plików

Zalety

- Duża szybkość działania

Wady

- Trudny w rozwoju
- Błąd = Kernel Panic

System plików w przestrzeni użytkownika

Zalety

- Łatwość tworzenia
- Możliwość pisania w wielu językach programowania
- Błąd = SIGKILL dla jednego procesu
- Duże możliwości

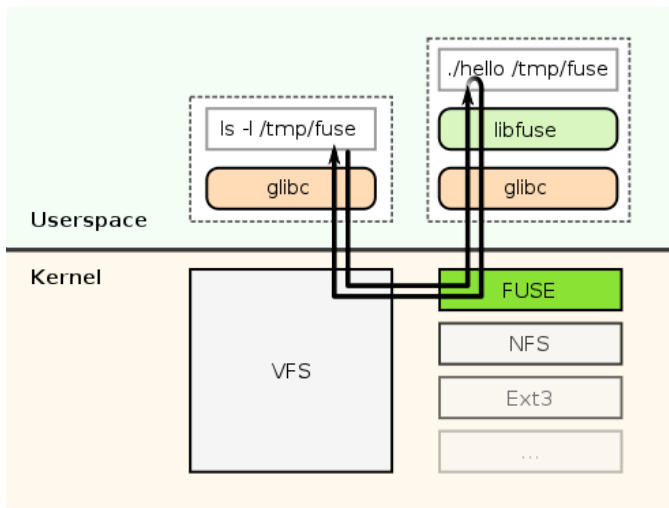
Wady

- Niska wydajność

Obsługiwane systemy

- Linux 2.4
- Linux 2.6
- FreeBSD (<http://fuse4bsd.creo.hu/>)
- NetBSD (puffs <http://www.netbsd.org/docs/puffs/>)
- Mac OS X (MacFUSE <http://code.google.com/p/macfuse/>)
- OpenSolaris (<http://www.opensolaris.org/os/project/fuse/>)
- GNU Hurd (<http://www.nongnu.org/hurdextras/#libfuse>)

Jak to działa?



Systemy plików korzystające z FUSE

ntfs-3g

- Pozwala na montowanie systemu plików NTFS w trybie do odczytu i zapisu
- Korzysta z wbudowanego modułu FUSE
- Całkowicie przezroczyste dla użytkownika
- <http://www.ntfs-3g.org/>

ZFS

- System plików stworzony przez Sun dla systemu Solaris
- Bezpieczeństwo, wydajność, mechanizm snapshotów...
- Niemożliwy do włączenia do kernela Linuksa ze względu na niezgodność licencji (GPL - CDDL)
- http://www.wizy.org/wiki/ZFS_on_FUSE

sshfs

- Pozwala na montowanie zdalnych zasobów poprzez ssh
- `sshfs user@zdalny_host:katalog/ /katalog/lokalny/`
- Dostępne wszystkie standardowe operacje na plikach
- Uploadowanie plików poprzez skopiowanie ich do punktu montowania
- <http://fuse.sourceforge.net/sshfs.html>

Offline FS

- Umożliwia łączenie zawartości wielu wymiennych nośników w jeden system plików
- Przy próbie odczytu pliku zostaniemy poproszeni o włożenie właściwego nośnika
- Można porządkować naszą offlinową bazę - tworzyć katalogi, przenosić pliki, itp
- <http://savannah.nongnu.org/projects/offlinefs/>

PicasaFS

- Pozwala na zarządzanie zdjęciami w serwisie PicasaWeb
- Umożliwia dostęp do informacji na temat tagów, opisu i innych metadanych zdjęcia
- <http://code.google.com/p/picasapush/>

Wiele innych..

- flickrfs - pozwala na montowanie zasobów flickra
- montowanie płyt audio i ich konwersja do formatów ogg/mp3
- GmailFS - montowanie konta Gmail
- WikipediaFS - montowanie stron opartych o silnik MediaWiki

Programowanie

Dostępne języki

- C (natywne API)
- C++
- Haskell
- TCL
- Python
- Perl
- Sh
- OCaml
- Ruby
- Lua
- Erlang
- PHP
- Java
- c#

Ruby!

Dlaczego?
Bo go nie znam... ;)

Podstawowe metody

- `contents(path)` - zwraca zawartość ścieżki `path`
- `directory?(path)` - zwraca `true`, jeśli `path` jest katalogiem
- `file?(path)` - zwraca `true`, jeśli `path` jest plikiem
- `read_file(path)` - zwraca zawartość pliku `path`

Najprostszy przykład

```
1 require 'fusefs'
2 class HelloDir
3   def contents(path)
4     [ 'hello.txt' ]
5   end
6   def file?(path)
7     path == '/hello.txt'
8   end
9   def read_file(path)
10    " Hello ,_World!\n"
11  end
12 end
13 hellodir = HelloDir.new
14 FuseFS.set_root( hellodir )
15 FuseFS.mount_under ARGV.shift
16 FuseFS.run
```

Metody: zawartość katalogu

- `contents(path)` - zwraca zawartość ścieżki `path`
- `directory?(path)` - zwraca `true`, jeśli `path` jest katalogiem
- `file?(path)` - zwraca `true`, jeśli `path` jest plikiem
- `executable?(path)` - `true`, jeśli `path` jest wykonywalny
- `size(path)` - zwraca rozmiar pliku dla ścieżki `path`

Metody: manipulacja plikami

- `read_file(path)` - zwraca zawartość pliku `path`
- `can_write?(path)` - zwraca `true`, jeśli do pliku można pisać
- `write_to(path, str)` - zapisuje `str` do plik `path`
- `can_delete?(path)` - zwraca `true`, jeśli plik można usunąć
- `delete(path)` - usuwa plik `path`

Metody: manipulacja katalogami

- `can_mkdir?(path)` - zwraca `true`, jeśli można utworzyć katalog
- `mkdir(path)` - tworzy katalog
- `can_rmdir?(path)` - zwraca `true`, jeśli można usunąć katalog
- `rmdir(path)` - usuwa katalog

psFS

Założenia:

- system plików wyświetlający procesy w systemie (jako katalogi)
- w każdym z katalogów plik „command” zawierający nazwę programu
- killowanie procesów poprzez usuwanie katalogów

Podsumowanie

Polecenia Linuksa są dość uniwersalne...

...i można nimi przedstawić praktycznie wszystko...

unzip; strip; touch; finger; mount; fsck; more; yes; umount; sleep;

Pytania?

- Strona FUSE: <http://fuse.sourceforge.net/>
- Strona CzLUGu: <http://czlug.icis.pcz.pl/>