FUSE – Filesystem in Userspace

FUSE – Filesystem in Userspace

- Základná myšlienka
  - umožniť implementáciu súborového systému v podobe bežného programu
  - modul v jadre systému zabezpečuje odovzdávanie požiadaviek z pripojeného FS príslušnému procesu
    - modul fuse
  - bežný proces zabezpečuje vykonávanie jednotlivých operácií

Na čo je FUSE dobré?

- umožňuje jednoduchšie vytváranie „ovládačov“ súborových systémov
  - tvorba bežných programov je jednoduchšia ako tvorba modulov do jadra
- umožňuje bežným používateľom pripájať si do stromu súborové systémy
  - bez toho, aby boli ich parametre fixne dané v /etc/fstab

Príklad použitia FUSE

- sshfs
  - umožňuje pripojiť súborový systém vzdialeného počítača prístupného protokolom SFTP
  - pripojenie

    sshfs meno@pocitac:cesta adresar
  - odpojenie

    fusermount -u adresar

FUSE – ako to funguje?

- modul fuse
  - zabezpečuje operácie na strane jadra
- libfuse
  - knižnica používaná procesom implementujúcim FS
- /dev/fuse
  - znakové zariadenie na komunikáciu medzi modulom fuse a procesom
- fusermount
  - pomocný program na (un)mount-ovanie

FUSE – ako to funguje?

- sshfs vytvorí spojenie so vzdialeným serverom
- do libfuse odovzdá smerníky na funkcie realizujúce jednotlivé operácie
- prostredníctvom libfuse spustí fusermount
- fusermount (set-uid) spôsobí pripojenie FS
- operácie v pripojenom FS sa cez /dev/fuse dostanú do sshfs
- sshfs vykoná cez SFTP operáciu a výsledok vráti

FUSE a práva

- proces, ktorý chce pripojiť FS na adresár musí mať do toho adresára plné práva
- fuse štandardne ignoruje kontrolovanie prístupových práv v pripojenom FS – ponecháva kontrolu na realizujúci proces
- voľbou **-o default_permissions** sa dá zapnúť štandardné kontrolovanie práv v jadre

FUSE a práva

- pripojený adresár je štandardne neprístupný pre iných používateľov ako toho, ktorý ho pripojil (ani pre root-a)
- voľbou **-o allow_other** môže **root** povoliť prístup aj iným používateľom
  - v konfiguračnom súbore /etc/fuse.conf je možné voľbou **user_allow_other** povoliť použitie **allow_other** a **allow_root** aj bežným používateľom (**allow_root** povoľuje prístup root-ovi)

Niektoré užitočné FUSE FS

- sshfs
  - sftp/ssh
- ntfsmount
  - nfts
- fusedav
  - WebDAV
- encfs
  - šifrovaný FS
- fuseiso
- fusesmb
  - SMB/CIFS
- gphotofs
  - PTP
- mtpfs
  - MTP
- obexfs
  - OBEX

Bootovanie v Linuxe

Proces bootovania

- BIOS načíta a spustí boot loader
- Boot loader načíta
  - jadro systému (kernel)
  - CPIO archív s obsahom prvotného koreňového FS (initramfs)
- Boot loader spustí jadro (a odovzdá mu parametre)
- jadro rozbalí initramfs do / a spustí **/init**
- **/init** pripojí skutočný koreňový FS a spustí z neho **/sbin/init**

Na čo initramfs?

- kedysi to fungovalo aj bez neho
  - jadro muselo obsahovať všetky potrebné ovládače na prístup ku koreňovému FS
- dnes je väčšinou jadro univerzálne a ovládače (v podobe modulov) sú v initramfs, odkiaľ sa pred pripojením skutočného koreňového FS zavedú
- to umožňuje podstatne komplikovanejší prístup ku koreňovému FS
  - RAID, LVM, šifrovanie, ...

Vytvorenie/údržba initramfs

- update-initramfs -u
  - aktualizuje initramfs pre aktuálne jadro
- update-initramfs -c -k verzia_jadra
  - vytvorí nový initramfs pre dané jadro
- konfigurácia
  - /etc/initramfs-tools/
    - initramfs.conf
    - modules
    - ...

Užitočné parametre jadra

- init=...
  - použije sa namiesto /sbin/init
- rdinit=...
  - použije sa namiesto /init
- ro, rw
  - či má byť koreňový FS pripojený ro alebo rw
- root=...
  - zariadenie, ktoré má byť použité ako koreňový FS
- číslo run-level-u / single / emergency

Užitočné parametre jadra

- root=
  - /dev/sda2
  - UUID=20282ab2-2692-4734-8806-f08e52171c0e
  - LABEL=root
  - /dev/nfs
    - nfsroot=[<server-ip>:]<root-dir>
    - ip=<client-ip>:<server-ip>:<gw>:<netmask>:<hostname>:<device>:<autoconf>

Ako BIOS číta boot loader?

- BIOS prečíta 1. sektor disku (MBR) a spustí ho
  - základ boot loader-a sa musí zmestiť do 446B
  - zvyšok sektora obsahuje partition table (4x16B)
- „klasický“ PC boot loader v MBR načíta 1. sektor **aktívnej** partície a spustí ho
  - ten buď načíta a spustí
    - jadro OS
    - alebo zvyšok väčšieho boot loader-a

Príklady boot loader-ov schopných spustiť Linux

- syslinux
  - FAT (napr. na USB kľúči)
- isolinux
  - ISO9660 (CD)
- pxelinux
  - boot zo siete (PXE)
- lilo
  - ext2/3 (+MBR)
- grub
  - rôzne (+MBR)

syslinux

- konfigurácia v súbore **syslinux.cfg**
- umožňuje zadávať aj parametre pre jadro
- umožňuje vytvoriť jednoduché boot menu

```
default inst

label inst
  kernel inst
  append initrd=instrd.gz
```

isolinux, pxelinux

- isolinux
  - konfigurácia v súbore **isolinux.cfg**
    - rovnaká ako syslinux
- pxelinux
  - konfigurácia v adresári **pxelinux.cfg**
    - rovnaká ako syslinux
  - načítava sa zo servera cez TFTP, server je určený pomocou DHCP
  - umožňuje nabootovať aj počítač bez disku
    - pre koreňový FS sa typicky použije nfs

lilo

- základ môže byť v MBR alebo v ext2/3 FS
- zvyšok v ext2/3
- pozíciu súborov si zapisuje v podobe čísel sektorov
  - problém vznikne, ak sa súbory presťahujú
- konfigurácia v **/etc/lilo.conf**
  - pri zmene treba preinštalovať
- inštalácia príkazom **lilo**

lilo

- umožňuje načítať a spustiť aj ďalší boot loader (chain loading)
  - umožňuje takto bootovať rôzne OS
- umožňuje vytvoriť boot menu
- umožňuje špecifikovať parametre pre jadro
- umožňuje modifikovať typ partície
  - užitočné na „skrývanie“ partícií

lilo

```
image = /boot/vmlinuz
  root = /dev/sda2
  label = Linux
  initrd = /boot/initrd.gz
  read-only

other=/dev/sda1
  label=windows
```

grub

- základ môže byť v MBR alebo v 1. sektore partície
- druhá časť môže byť medzi MBR a 1. partíciou alebo v partícii
  - keď je v partícii, je náchylná na zmenu umiestnenia
- konfigurácia a ďalšie časti (moduly) sú normálne vo FS
  - grub má moduly, ktoré mu umožňujú rozumieť rôznym FS (ext2/3/4, fat, ntfs, iso9660, ...)

grub

- umožňuje načítať a spustiť aj ďalší boot loader (chain loading)
  - umožňuje takto bootovať rôzne OS
- umožňuje vytvoriť boot menu
- umožňuje špecifikovať parametre pre jadro
- umožňuje modifikovať typ partície
  - užitočné na „skrývanie“ partícií

grub

- konfiguračný súbor **/boot/grub/grub.cfg**
  - generovaný príkazom **update-grub** alebo **grub-mkconfig**
  - parametre konfigurácie
    - /etc/default/grub
  - skripty používané pri generovaní
    - /etc/grub.d/

grub

```
menuentry 'Debian GNU/Linux, with Linux 3.2.0-0.bpo.3-686-pae'  {
        insmod part_msdos
        insmod ext2
        set root='(hd0,msdos3)'
        search --no-floppy --fs-uuid --set 20282ab2-2692-4734-8806-f08e52171c0e
        echo    'Loading Linux 3.2.0-0.bpo.3-686-pae ...'
        linux   /boot/vmlinuz-3.2.0-0.bpo.3-686-pae root=UUID=20282ab2-2692-4734-8806-
f08e52171c0e ro
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd.img-3.2.0-0.bpo.3-686-pae
}
```