

Administrace Unixu a sítě

```
inet6-adr: fe80::210:a4ff:fe01:9e5d/64 Rozsah:Linka
AKTIVOVÁNO VŠESMĚROVÉ_VYSÍLÁNÍ BĚŽÍ MULTICAST MTU:1500 Metrika:1
RX packets:66690 errors:0 dropped:0 overruns:0 frame:0
TX packets:100149 errors:0 dropped:0 overruns:0 carrier:0
kolizí:0 délka odchozí fronty:0
RX bytes:21490419 (20.4 MiB) TX bytes:10545763 (10.0 MiB)
```

4. Start systému, jádro

```
bug:/home/qiq# getent passwd | grep uucp
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
Debian-exim:x:102:102::/var/spool/exim4:/bin/false
qiq:x:1000:1000:Miroslav Spousta,2006,,:/home/qiq:/bin/bash
sshd:x:100:65534::/var/run/sshd:/bin/false
identd:x:101:65534::/var/run/identd:/bin/false
messagebus:x:103:104::/var/run/dbus:/bin/false
gdm:x:104:105:Gnome Display Manager:/var/lib/gdm:/bin/false
hal:x:106:106:Hardware abstraction layer,,:/var/run/hal:/bin/false
saned:x:109:109::/home/saned:/bin/false
bind:x:105:110::/var/cache/bind:/bin/false
smmta:x:107:111:Mail Transfer Agent,,:/var/lib/sendmail:/bin/false
smmsp:x:108:112:Mail Submission Program,,:/var/lib/sendmail:/bin/false
test:x:1001:1001:Test User,,:/home/test:/bin/bash
postfix:x:110:115::/var/spool/postfix:/bin/false
```

<http://www.ucw.cz/~qiq/vsfs/>

Xen

- virtuální servery (<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>)
- budete mít rootovská oprávnění ve virtuálním serveru
- přístup je po dobu výuky
- OS: Debian 3.1
- RAM: 32 MB, swap: 128 MB (/dev/sda2), root: 512 MB (/dev/sda1)
- několik síťových karet (eth0, eth1, ...)
- Síťové karty jsou propojeny virtuálními přepínači

Přístup k virtuálním serveru:

- **ssh login@kozel.vsfs.cz**
- **xencons localhost 90xx**
- xx je číslo serveru (login: **root**, heslo: žádné)

Start systému

- stejně jako ostatní OS je potřeba zavést do paměti jádro systému
- MBR, boot record: zavaděč OS
- v Linuxu: Grub, LILO, Solaris: OpenBoot
 - LILO je potřeba po každé změně v parametrech obnovit, Grub nikoli
- Grub přečte do paměti jádro systému, rozbalí ho, spustí ho
 - Grub může předat jádru parametry
- jádro po inicializaci připojí kořenový (root) filesystem a spustí program /sbin/init
 - jádro OS musí umět pracovat s kořenovým FS
 - často se používá ramdisk, kde jsou uloženy moduly pro všemožné FS
- Grub případně přečte ještě ramdisk, který může sloužit jako (dočasný) kořenový filesystem
 - Grub umí zavádnout kernel/ramdisk z RAID1

inittab

- místo initu je možné spustit jiný proces
 - např. v nouzi je možné použít parametr kernelu `init=/bin/bash`
- init přečte soubor `/etc/inittab` a podle něj řídí start systému
 - položky: identifikace:runlevel:akce:co_se_má_provést
 - akce: `respawn`, `wait`, `once`, `boot`, `bootwait`, `initdefault`, `sysinit`, `ctrlaltdel`
- runlevels: 0: halt, 1: singleuser, 2: multiuser (bez sítě), 3: multiuser, 5: multiuser s X, 6: reboot
- pro každý runlevel jsou definované služby, které mají běžet
- jsou v `/etc/init.d/`

```
id:2:initdefault:
si::sysinit:/etc/init.d/rcS
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
...
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
```

Run levels

- je definováno, které služby mají na kterém runlevelu běžet
- v `/etc/init.d/` jsou skripty pro spuštění a ukončení služeb
- v `/etc/rcX.d/` jsou symlinky do `/etc/init.d/` pro každý runlevel X
 - služby, které jsou nalinkovány `SYYslužba -> /etc/init.d/služba` mají na daném runlevelu běžet
 - `KYYslužba -> /etc/init.d/služba` naopak běžet na daném runlevelu nemají
- aktuální runlevel: **`who -r`**
- změna runlevelu: **`telinit runlevel`**
- restart systému: **`shutdown time [warning message]`**
- příkazy `reboot`, `halt`, `poweroff`
 - na Linuxu pokud není stav 0 nebo 6 volají `shutdown`, jinak provedou danou akci
 - na SunOS ale provedou akci hned

Linux (UNIX) jádro (kernel)

- bývá napsáno v C
 - možné optimalizace na CPU
- má za úkol spravovat
 - procesy (plánování, ochrana paměťového prostoru), někdy vlákna
 - meziprocesová komunikace (roury, síťová spojení)
 - signály, semaforey, sdílená paměť
 - virtuální paměť (správa, swappování)
 - souborové systémy (soubory, adresáře, jmenný prostor)
 - hardware (zařízení)
- možnosti nastavení
 - dynamické parametry (proc fs, sysctl)
 - parametry při spouštění jádra (v grub/LILO)
 - při kompilaci jádra

Modulární kernel

- části jádra jsou kompilovány jako speciální objekty (*.ko, á la .so)
- někdy je nutné (např. pro SunOS, nebylo možné překompilovat celé jádro)
- umožňuje za běhu měnit složení jádra a tím např. zabírat méně paměti
- manipulace s moduly (*.ko): **insmod**, **rmmmod**, **lsmod**
- mezi moduly mohou být závislosti
 - např. modul pro PCMCIA síťovou kartu vyžaduje ovladač PCMCIA rozhraní
 - příkaz **depmod** umí vytvořit soubor se závislostmi (**modules.dep**)
- příkaz **modprobe** zavede modul daného jména
 - na rozdíl od **insmod**, který vyžaduje kompletní cestu
 - zavolá ho kernel, pokud potřebuje nahrát modul pro nějaké zařízení
 - podle major a minor čísla zařízení se pozná, který modul je potřeba nahrát
 - případně je možné to upřesnit v **/etc/modules.conf**

Ovladače zařízení

- zařízení jsou identifikována podle major a minor čísla
 - major: typ zařízení – ovladač
 - minor: kolikátá instance daného zařízení
- znaková a bloková
- mknod, /dev/MAKEDEV (známe :-))
- do souboru /etc/modules.conf je možné přidat parametry pro ovladače daného zařízení
 - options io=0xXXXX irq=Y
 - případně napsat alias, který sváže označení (kernelové) s ovladačem daného zařízení
 - tak se dá např. říct, která síťová karta bude mít přiřazen určitý síťový interface (např. eth0, eth1)

sysctl, /proc

- parametry jádra je možné nastavovat pomocí sysctl(2) interface
 - /etc/sysctl.conf
 - nastavuje se při startu, nebo později pomocí příkazu sysctl(8)
 - závislé na systému, na SunOS: /etc/system
- /proc slouží ke zjišťování informací o procesech
 - každý běžící proces tam má svůj adresář s číslem PID
 - jsou v něm uloženy informace
 - otevřených souborech (fd/*), prostředí (environ)
 - mapa paměti (mem), spustitelná binárka, ...
- /proc/sys na Linuxu zpřístupňuje stejná nastavení jako sysctl
 - ale ve formě adresářů a souborů
 - je možné soubory číst a zapisovat do nich (zjištění a nastavení parametrů)
 - v /proc jsou další užitečné soubory (/proc/meminfo, ...)
 - ostatní UNIXy toto většinou nepodporují

Konfigurace jádra Linuxu (/proc)

- /proc:
 - cpuinfo, meminfo, vmstat, filesystems, swaps, devices, kcore, ...
- /proc/sys/
 - net/ipv4/ip_forward: povolení směrování paketů
 - net/ipv4/rp_filter: filtrování paketů z cizích sítí
 - net/ipv4/icmp_echo_ignore_broadcast: zakazuje ICMP reagovat na broadcasty
 - fs/file-max: max. počet otevřených souborů pro proces
 - vm/swappiness: jak moc se má swapovat (0 – vůbec, 100 – co nejvíc)
 - na úkor využití paměti jako cache
- /proc/acpi: ovládání ACPI (sleep, stav baterie, CPU throttling, ...)
- ...

Kompilace kernelu (Linux)

- proč kompilovat
 - nové ovladače, potřeba novějších vlastností
 - optimalizace (např. pro daný CPU)
 - vývoj ovladačů, modulů, samotného jádra
- jádro Linuxu je distribuováno ve formě zdrojových kódů
- obsahuje textový/konzolový/grafický konfigurační nástroj
 - pro nastavení požadovaných parametrů kompilace
 - make config, make menuconfig (vyžaduje curses), make xconfig (vyžaduje X11, Qt?)
- kompilace se spouští klasicky pomocí příkazu „make“
- „make install“ instaluje jádro (do rootu, případně do adresáře /boot)
- make modules_install instaluje moduly do /lib/modules/kernel-version

Debug věcí kolem kernelu (Linux)

- kernel má log (cyklický buffer)
 - sem vypisuje různá hlášení (co našel, co nenalezl, copyrighty :-), ...)
 - vypíše ho příkaz `dmesg`
 - případně se kopíruje do `/var/log/kern.log` a na konzoli
- magic `sysrq`
 - klávesová kombinace `alt-sysrq-něco`
 - musí být zapnuto (`/proc/sys/kernel/sysrq`)
 - umožňuje vypsat informace o aktuálním procesu, obsazení paměti, zabít procesy na terminálu, sync disků, ...
 - hodí se pro případ havárie
- při pádu systému: `Oops` + výpis `co` (proces) a `kde` (výpis zásobníku) zavinilo pád
 - systém někdy může dále běžet
 - `ksymoops` dekoduje výpis zásobníku pro zavedené moduly