
Zapínání a vypínání systému, proces `init`

Úvod do administrace operačních systémů

Jan Žďárek

České vysoké učení technické v Praze
FEL, katedra počítačů

(Program STM – kód Y36UAD)



Úvod

Komplexní (složitý) OS – komplexní (složitý) proces zapínání a vypínání.

Důvod: Zajištění správného pořadí, splnění závislostí.

- Bootování (bootstrapping),
- jádro,
- `init`,
- startup,
- používání OS,
- *runlevel* a jeho změna,
- vypínání (shutdown).



Bootování

Vstup: zapnutý HW (počítač).

Výstup: běžící počítač, spuštěn nějaký OS loader.

- Spuštění počítače (PROM, EEPROM, FEPROM),
- HW testy,
- zcela závislé na typu a výrobci počítače,
- musí podporovat minimálně načtení *superbloku* (boot bloku), kde je SW (OS *loader*), který umí načíst jádro z disku. Často bývá dokonce víceúrovňový.

PC – „BIOS“, primitivní (ale funkční) řešení.

Jiné počítače často mají mocnější nástroje:

SPARC – „OK prompt“, miniaturní OS v PROM (*monitor*).

– OpenBoot (IEEE 1275)



Bootování PC

Vstup: zapnutý PC HW.

Výstup: spuštěn nějaký OS loader.

PC BIOS – hledání a inicializace (některých) HW zařízení:

- operační paměť,
- klávesnice, porty, . . . , a
- zařízení, z něhož se má spouštět OS (BIOS: „boot device“).

Čtení *Master Boot Recordu* (1. sektor, 512 B, „boot sektor“),
tabulka rozdělení disku.

Zavaděč může číst další potřebné sektory se svým kódem (čtení fs).

Zavaděč zavede do paměti a předá řízení jádru OS nebo jinému zavaděči (chain loader).

LILO (Linux Loader) / GRUB (Grand Unified Boot Loader)



Bootování PC – LILO

LILO (Linux Loader)

- MBR / boot sektor,
- `/etc/lilo.conf`,
- konfigurace přímo v boot sektoru, nutné spustit `lilo`.

```
boot=/dev/hda
```

```
root=/dev/hda2
```

```
install=/boot/boot.b
```

```
map=/boot/map
```

```
delay=50
```

```
image=/boot/vmlinuz-2.6.22
```

```
label=Linux
```

```
read-only
```

```
image=/boot/vmlinuz-2.4.20
```

```
label=Zaloha
```

```
read-only
```



Bootování PC – GRUB

GRUB (Grand Unified Boot Loader) – komplexnější než LILO, „moderní“.

- MBR / boot sektor,
- `/boot/grub/*`; `/boot/grub/grub.conf`,
- konfigurace v souborech, není nutné vždy po změně konfigurace zapsat boot sektor. (Co se stane po zrušení partition s konfiguračními soubory?)

```
default=0
timeout=5
title Linux
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.22 ro root=/dev/hda2
title Zaloha
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.20 ro root=/dev/hda2
```



Bootování PC – podpora více OS

Podporuje LILO, GRUB, Win NT loader, ...

Rozdíly jsou jen v pohodlnosti konfigurace.

Postup instalace:

- nejprve vždy Windows, některé verze s oblibou přepíše MBR,
- některé verze Windows požadují bootovat z první partition,
- Win NT loader lze přemluvit k předání řízení jinému zavaděči, `C:\boot.ini`.

LILO:

```
other=/dev/hda1
label=MS W
table=/dev/hda
other=/dev/hda4
...
```

`C:\boot.ini`

`[Operating Systems]`

`...`

`C:\linboot.b="Linux boot sector"`



Startup – spuštění jádra

Vstup: Běžící zavaděč našel soubor s jádrem, umístil ho do RAM a předal mu řízení.

Výstup: Jádro běží a spustilo první proces (`init`).

Inicializace jádra

- test operační paměti,
- alokace reálné paměti pro struktury jádra,
- procesům je k dispozici zbytek.
- Detekce a inicializace HW.
- Vytvoření „procesů jádra“ (`fsflush` / `kflushd`, `kpiod`, `kswapd`, ...)
- Vytvoření prvního skutečného procesu – `init`.



Startup – spuštění systému

Vytvořením procesu `init` hlavní úloha jádra při startupu končí.

Následuje fáze nazývaná *startup*. Existuje několik standardních řešení:

1. AT&T System V:

- systém pracuje v tzv. *runlevelu*,
- skripty jsou organizovány v adresářích dle *runlevelů*.

2. BSD:

- jeden skript `/etc/rc` volá další a postupně inicializuje a spustí celý systém.

3. Solaris 10:

- nová koncepce, stejný cíl – nastartovat potřebné služby při splnění nutných závislostí (paralelní start, XML) `svc`.



Startup – spuštění systému

Vytvořením procesu `init` hlavní úloha jádra při startupu končí.

Vstup: Jádro běží a spustilo první proces (`init`).

Výstup: Systém je v požadovaném *runlevelu*.

Runlevel $\in \{0, 1, S, s, 2, 3, 4, 5, 6, \text{ale mohou být i další (obvykle se nepoužívají)}\}$.

Definovaný stav systému se spuštěnými/vypnutými službami (daemony). Administrátor může systém převést do libovolného runlevelu **R** příkazem

`# /sbin/telinit R; /sbin/init R.`

Konfigurace, spuštění/vypnutí služeb a zařízení pro každý runlevel: *rc-skripty*.



Startup – runlevel

runlevel	popis
0	přechod k vypnutému systému, Sol: PROM monitor
1	přechod do <i>single user módu</i> , Sol: administrative
S,s	single user mód = minimum procesů, údržba, opravy
2	multiuser, nonetwork
3	multiuser, network
4	normálně nepoužívaný
5	GNU/Linux: xdm (X11), Sol: power-off
6	restart systému
a,b,c	normálně nepoužívané
7,8,9	GNU/Linux, nepoužívané

Runlevely 0, 1 a 6 jsou rezervované.



Startup – rc-skripty (1)

► /etc/inittab

Formát: id:runlevels:action:process

(Gentoo) :

```
id:3:initdefault:          # Default runlevel.
si::sysinit:/sbin/rc sysinit # System init, mount local filesystems, etc
rc::bootwait:/sbin/rc boot  # Further system init, the boot runlevel.
l0:0:wait:/sbin/rc shutdown
l1:S1:wait:/sbin/rc single
l2:2:wait:/sbin/rc nonetwork
l3:3:wait:/sbin/rc default
l6:6:wait:/sbin/rc reboot
#z6:6:respawn:/sbin/sulogin
# TERMINALS
c1:12345:respawn:/sbin/agetty -f /etc/issue.logo 38400 tty1 linux
c2:2345:respawn:/sbin/agetty 38400 tty2 linux
# SERIAL CONSOLES
#s0:12345:respawn:/sbin/agetty 9600 ttyS0 vt100
ca:12345:ctrlaltdel:/sbin/shutdown -r now
pf:12345:powerfail:/sbin/shutdown -h now "Power Failure"
x:a:once:/etc/X11/startDM.sh
```



Startup – rc-skripty (2)

RC-skripty:

▶ `/etc/init.d/*; /etc/rc.d/*; ...`

Konfigurace:

▶ `/etc/sysconfig/*; /etc/conf.d/*; ...`

Identifikace a zamykání:

▶ `/var/lock/* (touch files); /var/run/*
(pids); ...`

Většina GNU distribucí má potřebu definovat svůj rc-systém trochu jinak, a umístit potřebné soubory trochu jinde než ostatní – malé, ale podstatné rozdíly.

▶ Princip fungování ale zůstává stejný – AT&T SV.



Singleuser/Multiuser mode

UNIXový systém může běžet a vykonávat užitečnou činnost zcela samostatně, bez jakéhokoliv GUI nebo jiné možnosti interaktivní práce uživatelů.

- Chceme-li počítač používat (lokální text/grafika), musíme nechat spustit program poslouchající na *terminálu* (`getty`; `xdm`; `gdm`).
+ podpora přihlašování (`login`; `sulogin`).
- Vzdáleně – `sshd`.
- V singleuser módu typicky běží jen jeden program `getty` na jednom (virtuálním) terminálu (`sulogin`).
- V multiuser módu typicky běží více `getty` a případně i `xdm` (X11).



Restart systému/Shutdown

Některé OS považují restart systému za běžný způsob, jak řešit problémy.

Dobře nakonfigurovaný UNIXový systém není potřeba restartovat.

(Některé systémy dokonce umí vyměnit i jádro OS bez restartu.)

Pokud to potřeba je, systém není vhodné vypnout „násilně“ (ztráta dat–cache, poškození fs).

- HW řešení: odpojení přívodu energie; CPU reset pin.

- SW řešení:

- `telinit`,
- `shutdown`,
- `halt`; `reboot`,
- `poweroff`.



Proces *init*

▶ `/sbin/init`:

- Na všech UNIXech je to první skutečný proces vytvořený po startu jádra,
- $PID \equiv 1$,
- je to proces, lze mu posílat signály (`# kill -HUP 1`),
- jeho prostřednictvím lze požádat o změnu *RL*,
`# /sbin/telinit 6`.
- Má významné postavení mezi procesy:
 - prarodič všech procesů v systému,
 - adoptuje osiřelé procesy,
 - adoptuje a zabraňuje uváznutí procesů v osiřelých skupinách procesů.