Linux Kernel – Historie a současnost

Martin Mareš <mj@atrey.karlin.mff.cuni.cz>, březen 1998

1. Struktura jádra OS

- Funkce jádra typického OS (mezičlánek spojující hardware a software, drivery, správa paměti a procesů).
- Monolitické (tradiční UNIXy), modulární (mikrokernelové systémy, AmigaOS) a pseudomodulární (Linux) architektury.
- Výhody a nevýhody jednotlivých architektur (monolity: obtížná rozšiřitelnost a malá portabilita, lze maximálně optimalizovat na rychlost, paměťová neúspornost; mikrokernely: často nepříliš rychlé, zato snadno distribuovatelné, výborně rozšiřitelné; pseudomodulární architektura: pokouší se spojit rychlost monolitu s rozšiřitelností modulárního OS).

2. Historie Linuxu

- Pravidlo sudých (stabilních) a lichých (vývojových) verzí

Verze	Čas	Řádky	Byty	Novinky
1.0	3/1994	176 250	4 719 743	První stabilní verze
1.2	3/1995	310 950	8 446 302	
2.0	6/1996	777 956	22 256 446	Více architektur, náznaky SMP
2.0.32	11/1997	893 560	25 400 390	
2.1.90	3/1998	1 451 411	41 548 428	Rozumný SMP, nový networking

- Linux over microkernel: MKLinux (OSF/Mach), Linux/L4, Linux/T4.

3. Pohled pod pokličku

- Rozdělení na architekturálně závislou a generickou část. Multiarchitekturální části (PCI apod.).
- Správa procesů a scheduler.
- Memory management, swapping, file mapping.
- Drivery: char, block, scsi, cdrom, net, isdn, fc4, video, pci, sbus...
- Filesystemy: generická část + jednotlivé FS: ext2, msdos, nfs, proc...
- Síťová vrstva: IPv4, IPv6, UNIX, IPX... (též packet scheduling apod.)
- A kde že vlastně ten kernel je? (`/boot/`, `vmlinuz` a `System.map`)

4. Kernelové moduly

- Princip činnosti kernelových modulů (vlastně dynamické linkování jádra).
- Příkazy `insmod`, `rmmod` a `lsmod`.
- Automatické instalování a deinstalování modulů: `depmod`, `modprobe`, `kerneld` a `kmod`.
- `/etc/conf.modules` a `/lib/modules/...`

5. Aplikace z pohledu kernelu
   - Procesy (se sdílenými či nesdílenými prostředky). Stavy procesů (ready, waiting, zombie) a přechody mezi nimi.
   - Adresní prostor (text, data, stack, shared text, kernel space).
   - File handles.
   - Signal handlers.
   - IPC: fronty, semafory a sdílená paměť.
6. Kernel z pohledu aplikací
   - Systémová volání (jak nativní, tak emulovaná)
   - Sdílené knihovny a dynamický linker.
7. Systémová volání
   - Všechno jsou soubory: normální soubory, adresáře, char/block devices, pipe, named pipe, socket. Blocking a non-blocking mód. File handles a jejich číslování (`close`/`dup` trik na redirekci).
   - Ne všechno jsou soubory: IPC.
   - Zakládání procesů: `fork`, `exec`, `exit` a `clone`.
   - Aktivní a neaktivní čekání (`pause`, blocking, `select`).
   - Ovládání signálů (`signal`, `sigaction`, `kill`, `alarm`).