Linux
Teorie operačních systémů a realita

České vysoké učení technické Fakulta elektrotechnická

Historie Unixu a Linuxu

MULTICS

- 50. - 60. léta minulého století, osobní počítače ve smyslu užití.
- Rozměrově však potřebovaly několik místností
- MULTiplexed Inforamtion and Computing Service

UNICS -> UNIX

- zjednodušená verze MULTICSu pro PDP-7

PDP-11 UNIX

- 70. léta
- B -> C

Berkeley UNIX

- 1BSD

Standard UNIX

- System V, POSIX
- konec 80. let

MINIX

- 1987

Historie Unixu a Linuxu

Linux

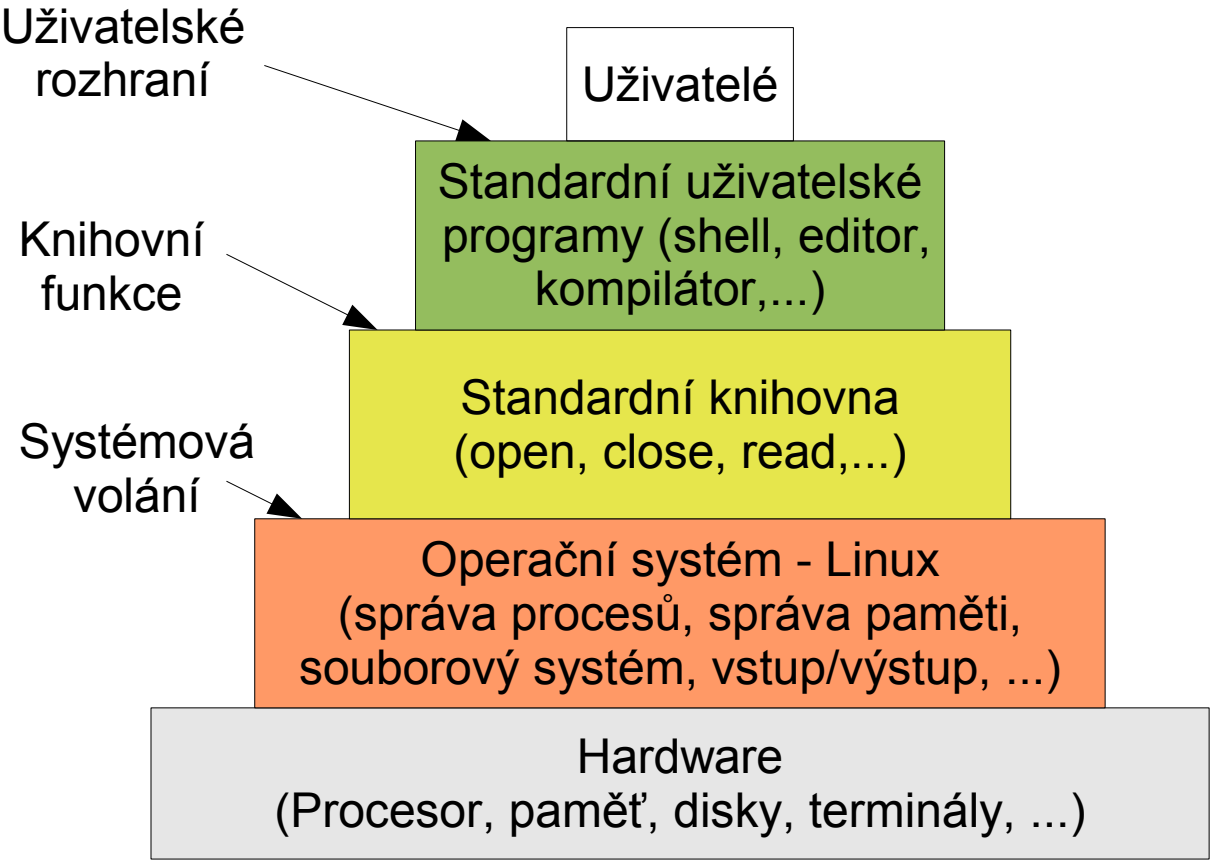
- verze 0.01 - 1991
- velké množství distribucí: *www.distrowatch.com*
- FreeBSD

Přehled Linuxu

Určení Linuxu

- víceuživatelsklý a víceprocesorový systém
- zaměřený na znalé uživatele

Rozhraní v Linuxu:

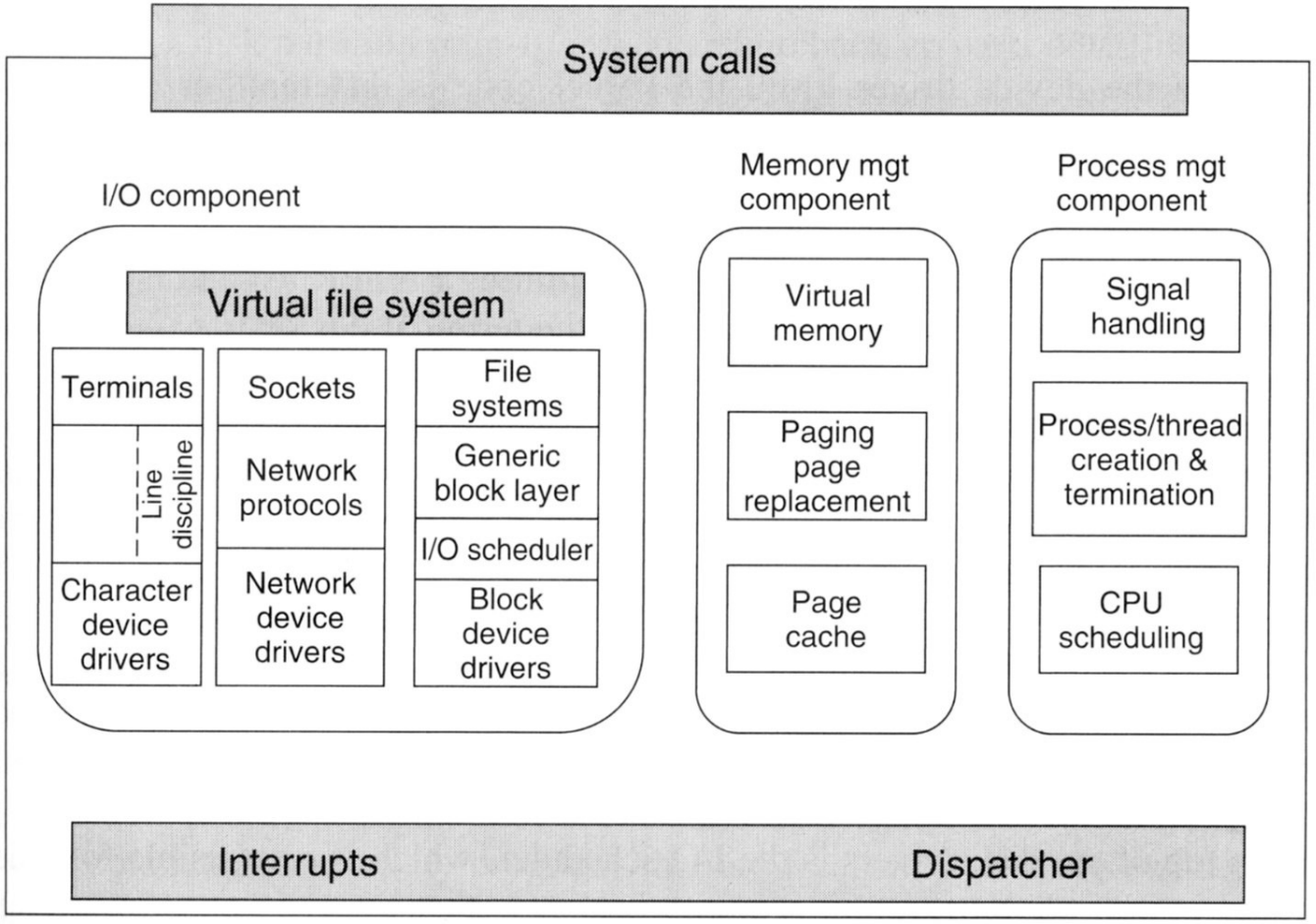


Přehled Linuxu

Shell + Utility

- nenáročný na zdroje (monitor + klávesnice)
- velmi výkonný a flexibilní
- seskupováním jednotlivých jednoúčelových nástrojů lze dosáhnout požadované výsledky
- skriptování
- utility: cat, chmod, cp, cut, grep, head, ls, make, mkdir, paste, ps, rm, rmdir, sort, tail, tr, ...

Přehled Linuxu - Struktura jádra



Procesy v Linuxu

Koncept procesů v Linuxu

- proces (úloha) aktivní entita
- proces provádí jeden program a na počátku má jen jedno vlákno
- víceúlohový systém
- proces má vlastníka (nemusí se jednat o fyzicky existujícího uživatele - démoni)
- vztahy mezi procesy: rodič - potomek (PID, fork)
- komunikace mezi procesy: roury (pipe) a signály
  POSIX: SIGABRT, SIGALRM, SIGFPE, SIGHUP, SIGILL, SIGQUIT, SIGKILL, SIGPIPE, SIGSEGV, SIGTERM, SIGUSR1, SIGUSR2

Procesy v Linuxu

```
main ()
{ ...
  pid = fork();
  switch (pid) {
  case -1: /* doslo k chybe */
    perror ("chyba ve funkci fork()");
    exit(1);
  case 0: /* program provadeny v potomkovi */
    printf ("PID procesu potomka: %d\n", (int) getpid ());
    execlp("sleep", "sleep", "30", (char *) NULL);
    perror ("chyba ve funkci execlp()");
   exit (1);
  default: /* program provadeny v rodici */
    printf ("PID procesu rodice : %d\n", (int) getpid ());
    wait(&status);
  };
...
}
```

Procesy v Linuxu

Systémová volání správy procesů

Systémové volání	Popis
pid = fork()	vytvoří potomka identického s rodičem
pid = waitpid(pid, &statloc, opts)	čeká na ukončení činnosti potomka
s = execve(name, argv, envp)	nahradí kód volajícího procesu jiným
exit(status)	ukončení provádění s návratovou hodnotou
s = sigaction(sig, &act, &oldact)	definice akce při přijmu signálu
s = sigreturn(&context)	návrat z obsluhy signálu
s = sigprocmask(how, &set, &old)	manipulace s mskováním signálů
s = sigpending(set)	vrací množinu blokovaných signálů
s = sigsuspend(sigmask)	změna masky signálu a suspendování procesu
s = kill(pid, sig)	zaslání signálu procesu
residual = alarm(seconds)	nastavení alarmu
s = pause()	suspendování procesu až do příchodu signálu

Procesy v Linuxu

- Implementace procesů a vláken.
  - Proces nebo vlákno, všechno je úloha:
  - datová struktura *task_struct*
    - Parametry pro plánování
    - Ukazatele do paměti, případně na disk s odloženými stránkami paměti
    - Signály
    - Registry
    - Stav systémových volání
    - Tabulka deskriptorů souborů
    - Účetnictví
    - Zásobník pro běh v režimu jádra
    - Různé

Procesy v Linuxu

Plánování:

1) Real-time FIFO

2) Real-time round robin

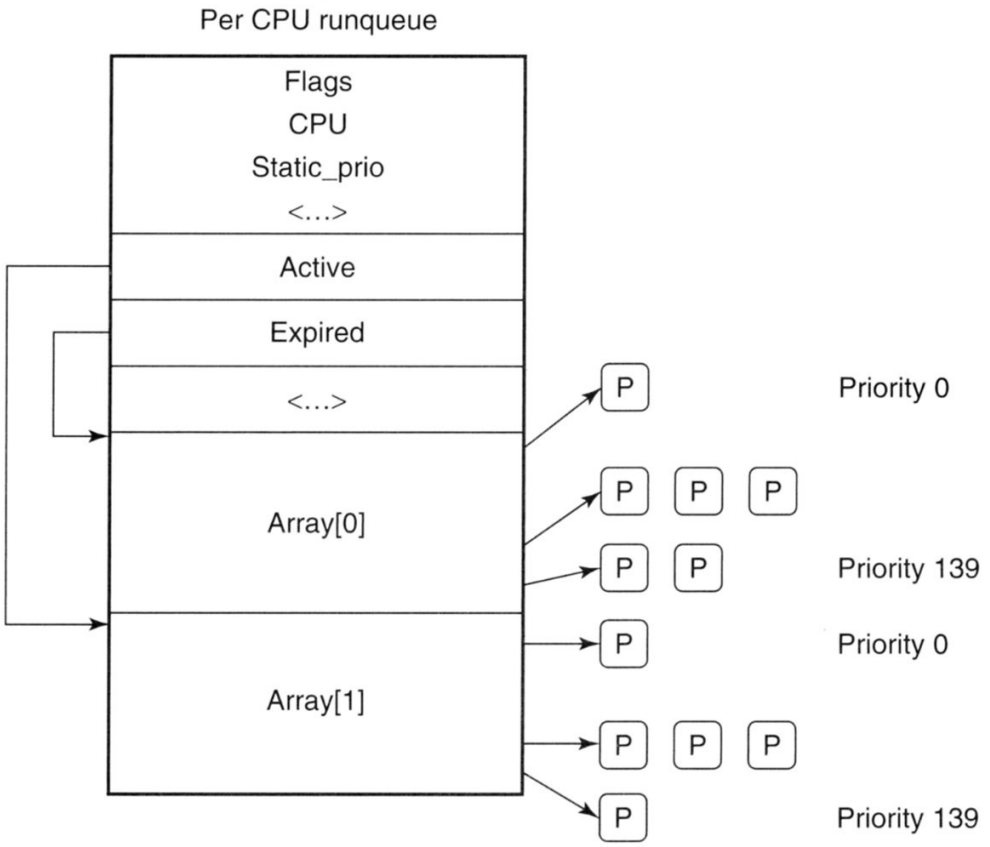
3) Timesharing

-statické priority (nice) -20 - +1

Zavádění systému - Boot

- kde se vzal první proces?

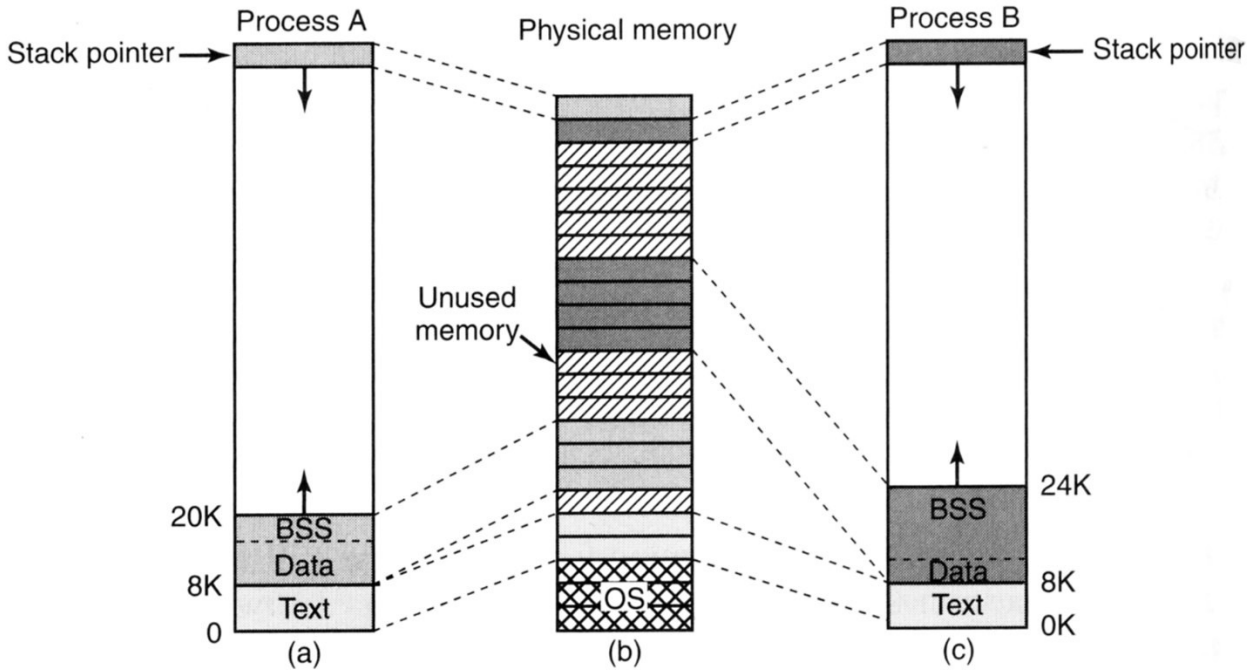
- GRUB vs. LILO



Správa paměti v Linuxu

Koncept: tři segmenty na proces (text, data, stack)

- sdílené segmenty kódu, soubory mapované do paměti



Správa paměti v Linuxu

Systémová volání správy paměti
- nejsou standardizována normou POSIX
- u přenositelných programů je třeba se spoléhat na knihovní funkce (malloc)

Implementace správy paměti
- maximum paměti pro proces obvykle 3GB + 1GB navíc v módu jádra

Fyzická paměť:
- DMA
- Normal
- Highmem

Stránkování
- čtyř úrovňová tabulka stránek

Vstup a výstup v Linuxu

Koncept
- speciální soubory v adresáři /dev
- blokově a znakově orientovaná zařízení
- major, minor number

Sítě
- socket
- TCP, UDP

Systémová volání
Implementace
- ovladače (drivery)
- přímo v jádře nebo moduly

Souborové systémy v Linuxu

Koncept

- původně souborový systém MINIX 1 (jména 14 znaků, soubor max. 64 MB)
- soubor: libovolná sekvence bytů (třeba i nulové délky)
- nerozlišují se binární a ASCII soubory
- absolutní a relativní cesty
- odkazy (link)
- zámky

Systémová volání: create, open, close, read, write, stat, fcntl

Implementace

- VFS
- Ext2, Ext3, Ext4
- NFS

Bezpečnost v Linuxu

Koncept

- UID, GID
- u souborů i procesů
- práva: číst, psát, spouštět
- ACL

Systémová volání: chmod, access, getuid, geteuid, getgid, getegid, chown, setuid, setgid

Implementace:

- autentifikace - login
- login zajistí běh uživatelových procesů se správným uid a gid

Bezpečnost v Linuxu