

UNIX

Historie a základní pojmy

Historie

Před ním

- Multics (Multiplexed Information and Computing Service) 1965-2000
 - MIT Project MAC
 - Multiple Access Computer (Corbató)
 - Man and Computer (Minsky)
 - Bell Telephone Laboratories – BTL, výzkumná složka spoluvlastněná AT&T
 - General Electric
- 10/69 GE645
- 4/69 BTL odstupuje od projektu
- shell, příkazy ls, pwd, cd/cwd, argumenty, skripty

Začátek

1969

**Computing Science Research Center - CSRC (BTL)
zůstalo bez interaktivního prostředí (OS)**

**Ken Thompson, Denis Ritchie, ... návrh systému souborů
→ UNIX**

Thompson

- napsal simulační program pro systém souborů a stránkování na žádost (v Multicsu)
- napsal jednoduché jádro pro GE 645
- napsal „Space Travel“ ve Fortranu pro GECOS (OS pro GE), neuspokojivé
- našel PDP7 (Digital Equipment Corporation)
- s Ritchiem přepsali „Space Travel“, křížový překladač na GECOSu, děrné pásy, naučili se vytvářet programy pro PDP-7

Thompson a Ritchie implementují

- návrh systému souborů, procesy
- systémové programy copy, print, delete
- jednoduchý shell

1970

- Brian Kernighan
slovní hříčkou vytváří název Multics – Unics – Unix
- pro patentové oddělení BTL, příprava textů, PDP 11
- Thompson píše překladač pro FORTRAN, ovlivněn BCPL - Basic Combined Programming Language
vzniká jazyk B, interpret

1971

- UNIX PROGRAMMER'S MANUAL
- UNIX Time-Sharing System First Edition (V1)
 - o 60 příkazů, b (přelož b program), cat, chmod,...
- první instalace v patentovém oddělení BTL

1972

- Ritchie vytváří jazyk C
- McIlroy navrhuje rouru (*pipe*)

1973

- přepsán do C
- počet instalací v BTL ~25
- vytváří se UNIX Systems Group – USG
- publikace The UNIX Time Sharing System
 - o cm.bell-labs.com/cm/cs/who/dmr/CAM.html
- Fourth Edition (V4)

AT&T nesmí prodávat počítačové produkty, poskytuje licence (University of California at Berkeley)

Šíření

1975

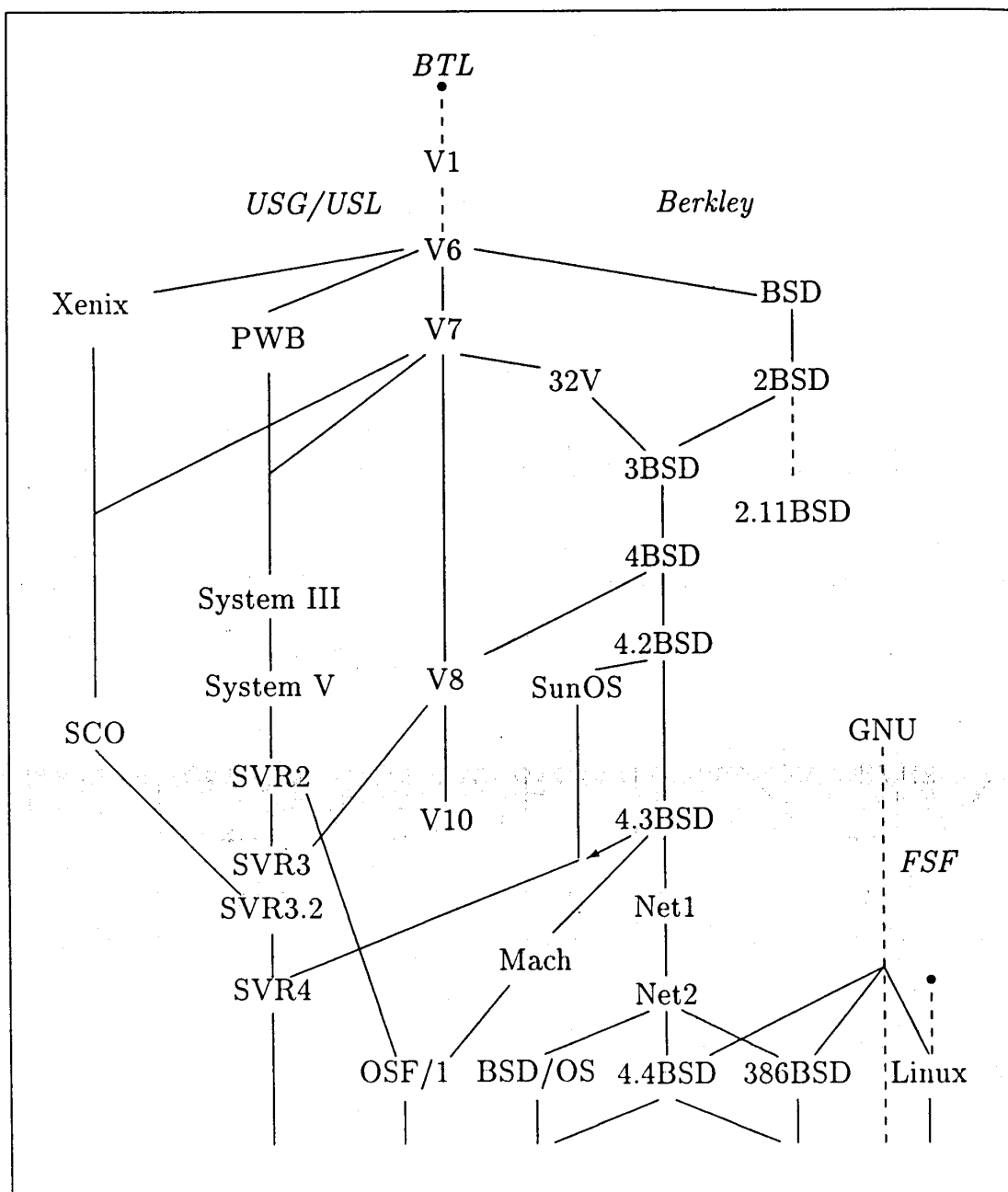
- Sixth Edition (V6)
 - o odvozena 1BSD – Berkeley Software Distribution
 - o PWB/UNIX – programmer's workbench (BTL)
 - o 1980 XENIX Microsoft + Santa Cruz Operation – SCO

1977

- přenos na jinou architekturu Interdata 8/32

1978

- přenos na 32 bitový VAX-11 (DEC) UNIX/32V



1979

- **Seventh Edition (V7)**
 - **Bourne Shell**

- **Vývoj**
- **Komerzializace**
- **Normy**

Vývoj

- **Berkeley BSD**
- **AT&T**
- **Carnegie-Mellon University**

Berkeley BSD

1978

- **2BSD C shell**
- **Berkeley získalo VAX-11 a UNIX/V32**

1979

- **3BSD virtuální paměť - stránkování**

1980 – 1993

- **4BSD**
- **1980 4.0BSD, 1981 4.1BSD, 1983 4.2BSD, 1986 4.3BSD, 1993 4.4 BSD**
- **TCP/IP, vi editor, sokety, ...**

AT&T

1982

- AT&T může vstoupit na počítačový trh
- UNIX System III

1983 – 1993

- System V
- 1983 System V, 1984 System V Release 2 (SVR2), 1987 SVR3, 1989 SVR4 (AT&T + Sun, SVR3 + 4BSD + XENIX)
- 1989 AT&T vytváří UNIX Systems Laboratories – USL
- 1993 SVR4.2MP
- virtuální paměť – oblasti (*region*), prostředky meziprocesové komunikace, proudy (*streams*)
Korn shell

Carnegie-Mellon

1985 -

- Mach
- vzniká v 4.2BSD, kde Mach nahrazuje části jádra 4.2BSD
- Mach 3, BSD kód mimo jádra, první mikrojádru
- 4.3BSD, OSF/1 (DEC), ...

Komericializace

- **XENIX**
- **AIX (IBM), HP-UX (Hewlett-Packard), ULTRIX, OSF/1 (DEC), SunOS – 4BSD, Solaris – System V (Sun Microsystems)**
- **1988**
 - OSF – Open Software Foundation (DEC, IBM, HP), OSF/1**
 - UNIX International – UI (AT&T, Sun), System V**
- **1993**
 - UI končí, AT&T prodává celý UNIX Novellu**
 - Novell přenáší UNIX na X/Open**
 - Sun kupuje práva na SVR4 od Novellu**
- **1996**
 - OSF a X/Open vytváří Open Group**

Normy

- **AT&T vydává SVID – System V Interface Definition**
- **IEEE – Institute of Electrical and Electronics Engineers, www.ieee.org vydává POSIX – Portable Operating Systems based on UNIX**
 - POSIX.1 - 1988 systémové rozhraní**
 - POSIX.2 - 1992 uživatelské rozhraní (shell)**
- **X/Open konsorcium vydává X/Open Portability Guide – XPG, 1990 XPG3, 1993 XPG4, XPG4v2 – Single UNIX Specification, 1996 s OSF vytváří Open**

Group

1998 – 2001

sjednocení ☺

pracovní skupina (Austin Group)

- IEEE Portable Applications Standards Committee**
- Open Group**
- ISO/IEC JTC1/SC22/WG15**

International Organization for Standards, iso- = ISO

International Electrotechnical Commission

Joint Technical Committee: Information

Technology Standards

Subcommittee 22: PROGRAMMING

LANGUAGES, THEIR ENVIRONMENTS

AND SYSTEM SOFTWARE INTERFACES

Working Group 15: POSIX

IEEE – IEEE Std 1003.1-2001 (POSIX.1)

Open Group – Base Specifications, Issue 6

jádro Single UNIX Specification, Version 3

ISO/IEC –

Struktura:

- Base Definitions, Issue 6 (XBD)**
- Shell and Utilities, Issue 6 (XCU)**
- System Interfaces, Issue 6 (XSH)**
- Rationale (Informative) (XRAT)**

LINUX

Před ním

1983

- **Richard Stallman oznamuje začátek projektu GNU – GNU is Not Unix (rekurzivní akronym)**

1984

- **odchází z MIT, má povolení používat zařízení AI laboratoře**
- **vytváří GNU C Compiler – GCC a GNU Emacs**
- **softvér je free = svobodný**

1985

- **vzniká Free Software Foundation – FSF, pro GNU**

1987

- **Andrew Tanenbaum MINIX 1.0, 4.77MHz, 256KB RAM, 360KB pružný disk**

Začátek

1991

- **Linus Torvald**
přenáší na 386PC GCC a BASH (Bourne Again Shell)
- **GNU nemá OS (HURD, Mach)**

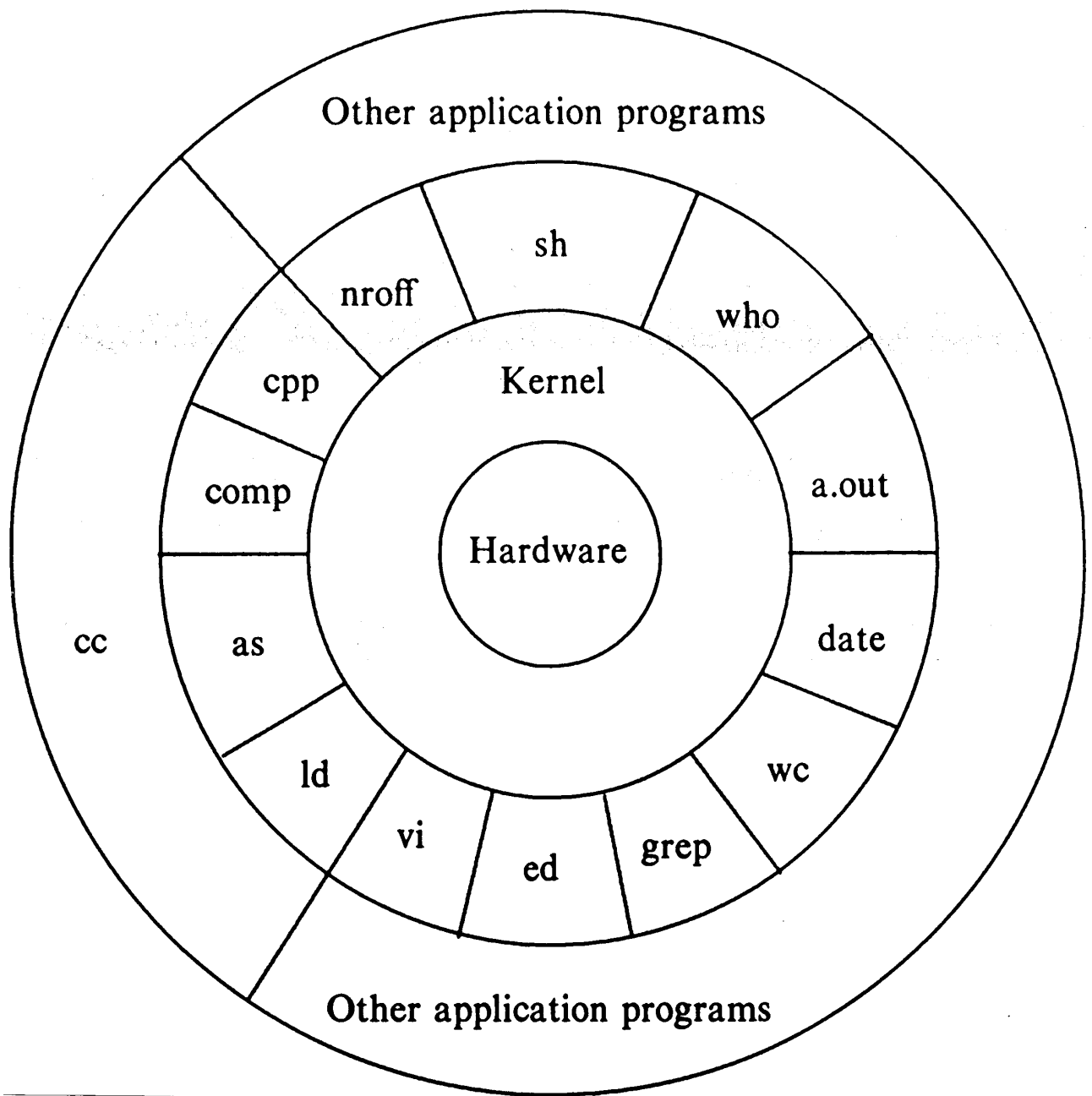
Šíření

HW

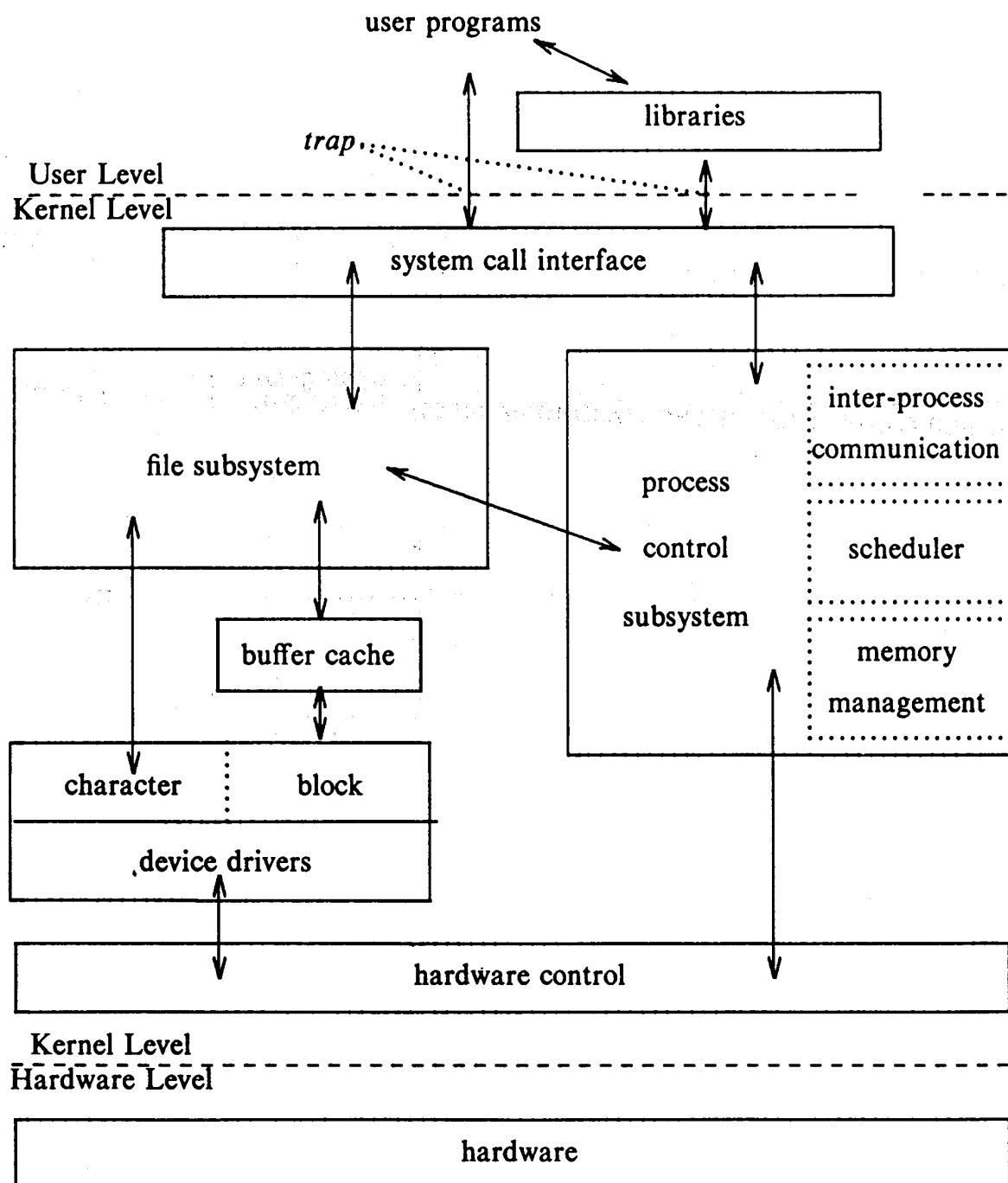
- **Acorn, Compaq Alpha, Intel 80x86, Motorola MC680x0, MIPS, Power PC, SPARC, Ultra Sparc – 64 bitů, IBM System/390**
- **IBM podporuje Red Hat Linux Advanced Server**
- **UnitedLinux vytvořily Caldera, SuSE, ...**

Základní pojmy

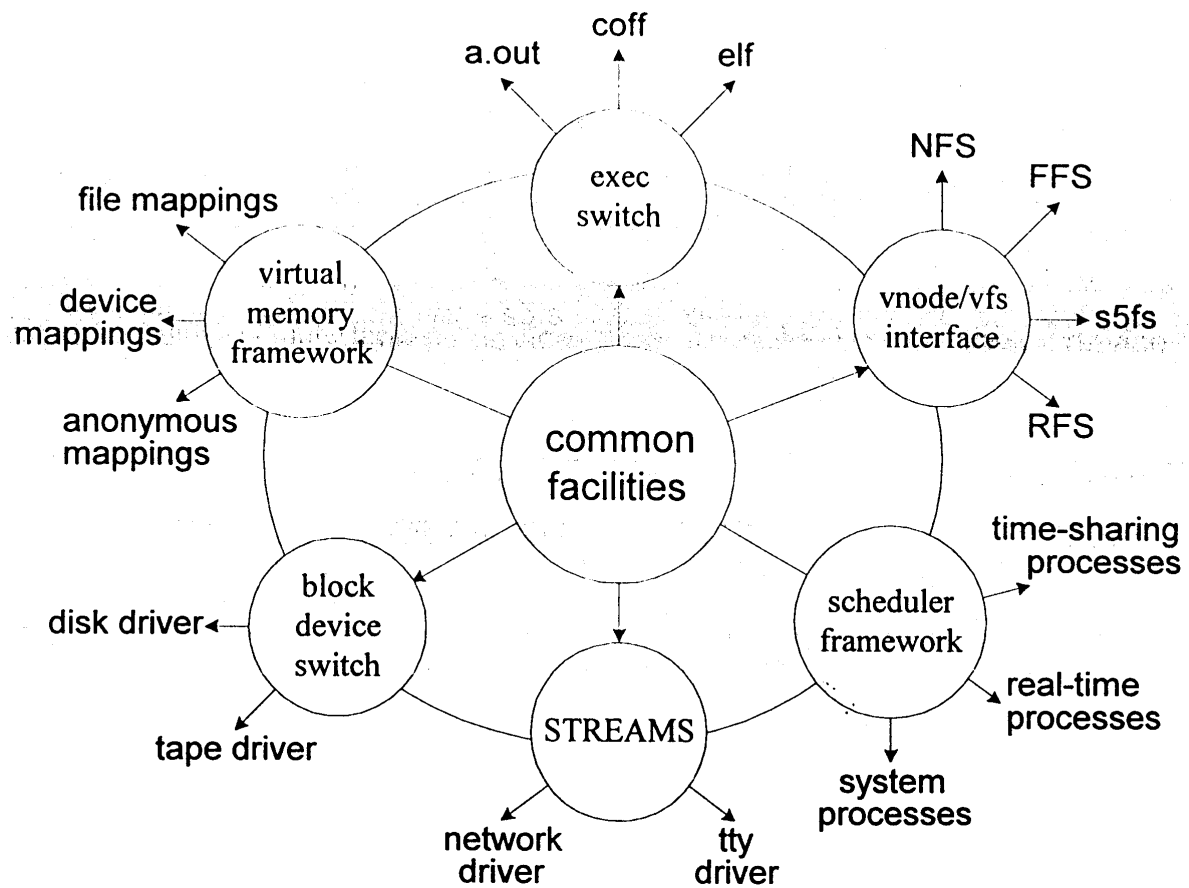
Jádro OS – kernel



Struktura jádra (tradiční)



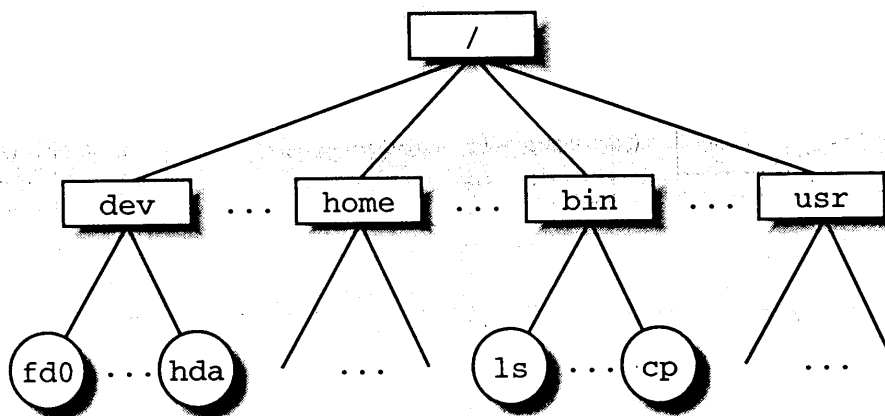
Struktura jádra (moderní)



Vahalia 96

Systém souborů

- hierarchická struktura
- organizován jako strom (acyklický graf -
 `ln zdrojový_soubor cílový_soubor`)
- listy jsou soubory (prázdné adresáře)
- ostatní vrcholy adresáře (*directory*)
- jméno souboru nebo adresáře ::= řetězec_znaků
- jméno kořenového adresáře /
- okamžitý pracovní adresář (*current working directory*)
- určení souboru (adresáře) cesta
 - o absolutní – od kořene
 - o relativní – od pracovního adresáře
- „.“ označuje pracovní adresář
- „..“ označuje rodičovský adresář



Typy souborů

- obvyčejné (*regular*)
- adresáře (adresářové soubory)
- symbolické odkazy
- blokově orientované soubory zařízení (*device*)
- znakově orientované soubory zařízení
- roury (*pipe*) a pojmenované roury (FIFO)
- sokety

Soubory neobsahují informace o souboru (typ, délka, ...)

Adresáře slouží na organizaci hierarchie a přístup pomocí jmen

Všechny informace o souboru jsou v i-uzle (*inode*, *index node*)

V adresáři se nachází dvojice (jméno, číslo i-uzlu) – odkaz (*file hard link*)

Symbolický odkaz – soubor obsahující cestu k souboru (*soft, symbolic link*)

```
ln -s zdrojový_soubor cílový_soubor
```

Přístupová práva

- vlastník (*owner*)
- skupina (*group*)
- ostatní (*others*)

- **rwx** read, write, execute

Systémová volání souborového systému

- procesy pracují jenom s otevřenými soubory

```
fd=open(cesta, příznak, mód);  
                flag           mode
```

fd – deskriptor souboru

cesta – specifikuje soubor

příznak – specifikuje jak má být soubor otevřen
(čtení, zápis, vytvoření...)

mód – specifikuje přístupová práva vytvářeného
souboru

```
fd = creat(cesta, mód);
```

- čtení a zápis

```
přečteno = read(fd, buf, počet);
```

```
zapsáno = write(fd, buf, počet);
```

Příklad

```
char b[1024];
```

```
main(int argc, char *argv[])
```



```

{
    int fds, fdn;
    mode_t mod = S_IRUSR | S_IWUSR |
                 S_IRGRP | S_IROTH;

    if (argc != 3)
    {
        printf("musi byt 2 argumenty\n");
        exit(1);
    }
    fds=open(argv[1], O_RDONLY);
    if (fds == -1)
    {
        printf("neotevren soubor %s\n",
               argv[1]);
        exit(1);
    }
    fdn=creat(argv[2],mod);
    if (fdn == -1)
    {
        printf("nevytvoren soubor ");
        printf("%s\n",argv[2]);
        exit(1);
    }
    kopiruj(fds, fdn);
    exit(1);
}

```

```

kopiruj(int s,int n)
{
    int pocet;

```

```
while((pocet =  
        read(s, b, sizeof(b))) > 0)  
    write(n, b, pocet);  
}
```

Standardní soubory

standardní vstupní soubor	fd	0
standardní výstupní soubor	fd	1
standardní chybový soubor	fd	2

další přidělený deskriptor souboru bude 3, ...

- zavření souboru

```
close(fd);
```

fd je uvolněn

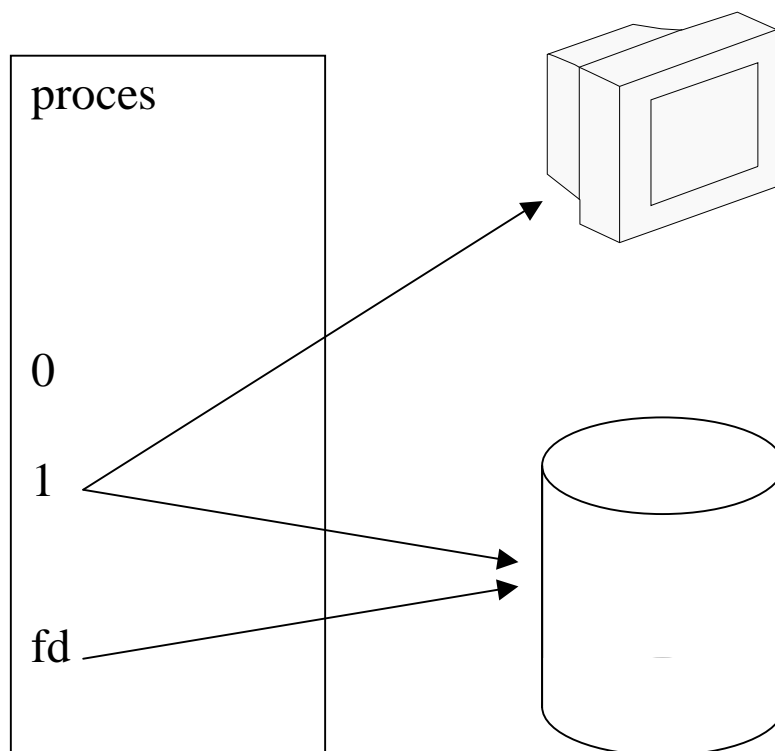
Přesměrování V/V

```
novýfd = dup(fd);
```

- zdvojí deskriptor souboru
- soubor je přístupný oběma deskriptory
- novýfd je nejmenší, který je k dispozici

Přesměrování standardního výstupního souboru

```
...  
int fd;  
...  
close(1);  
dup(fd);  
close(fd);  
...
```



Tradiční systém souborů

disk

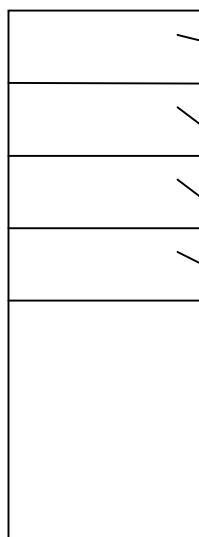
bootovací blok	super blok	seznam i-uzlů	údajové bloky
-------------------	---------------	------------------	------------------

obyčejné soubory,
adresáře, ...

- bootovací blok může obsahovat kód pro zavedení systému
- super blok obsahuje informace o stavu souborového systému
- seznam i-uzlů, informace o jednotlivých souborech

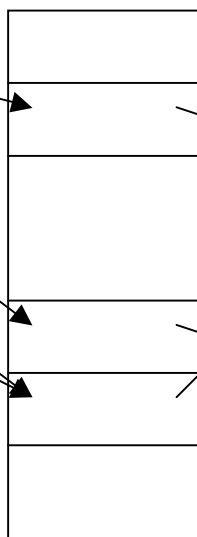
jádro

tabulka deskriptorů
souborů



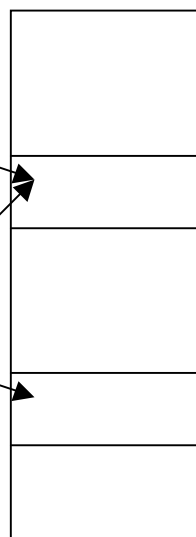
každý proces

tabulka
souborů



jedna

tabulka
i-uzlů



jedna

tabulka deskriptorů souborů

- deskriptor souboru index do tabulky
- položka obsahuje ukazatel do tabulky souborů

tabulka souborů

- obsahuje údaje pro jednotlivá otevření souboru
 - o pozici (*offset*) v souboru pro další čtení nebo zápis
 - o přístupová práva pro proces

tabulka i-uzlů

- obsahuje údaje z i-uzlu na disku a další, např. počet odkazů

adresář

27	muj
50	tvuj

číslo jméno
i-uzlu souboru

```
fd1=open("muj",...);
```

```
fd2=open("tvuj",...);
```

```
fd3=open("muj",...);
```

```
fd4=dup(fd3);
```

