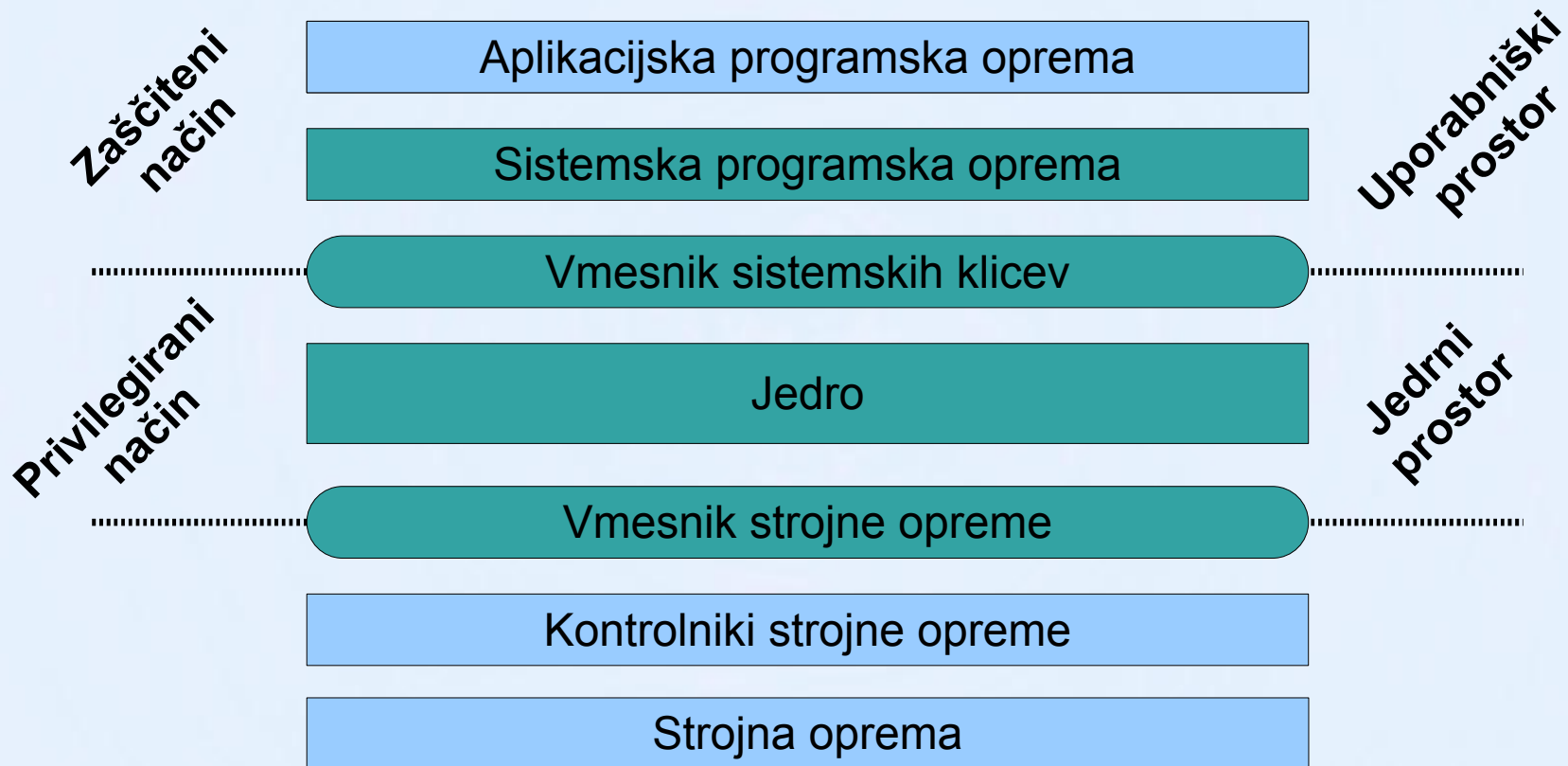


Jedro Linux



Računalniški sistem



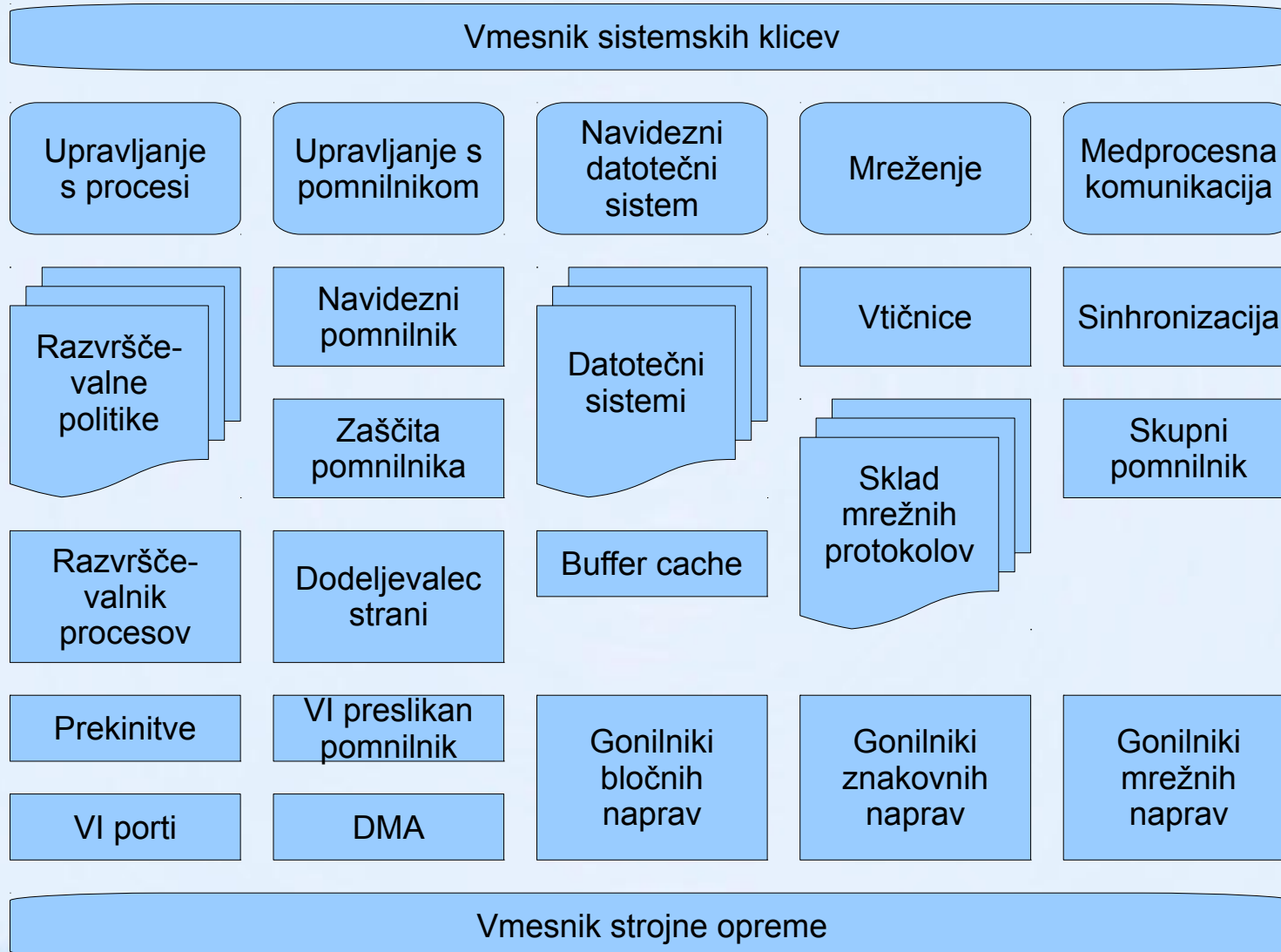
Vmesnika

- Vmesnik sistemskih klicev.
 - Komunikacija med uporabniškimi programi in jedrom.
 - Vstop v jedro preko sistemskih vrat (*system gate*).
- Vmesnik strojne opreme.
 - Komunikacija med jedrom in strojno opremo.
 - VI preslikan pomnilnik.
 - VI porti – ločen naslovni prostor, ukaza in in out.
 - DMA – neposreden dostop do pomnilnika.
 - Strojne prekinitve.

Jedro

- Jedrni prostor / privilegirani način.
 - Jedro teče v privilegiranem načinu delovanja CPE.
 - Omogočen dostop do celotnega pomnilnika.
 - Linearni naslovni prostor je enak fizičnemu.
 - Omogočen dostop do strojne opreme.
 - Omogočeno izvajanje privilegiranih ukazov.
 - `sti`, `cli`, `int`, itd.

Struktura jedra



Jedro

- Monolitno jedro.
 - Velik kos kode, ki vsebuje *skoraj vse*.
 - Sprememba izvorne kode → ponovno prevajanje.
 - Hitrost (enotni naslovni prostor, manj preklapljanja).
- Modularno jedro – LKM (*loadable kernel modules*).
 - Modul (kos prevedene kode) je možno vložiti (in odstraniti) v jedro tekom samega izvajanja.
 - Namenjeni za gonilnike naprav, datotečnih sistemov, mrežnih kartic itd.

Jedro

- Osnovni podatki o jedru: `uname`.
- Jedrna obvestila: `dmesg`.
- *Nastavljanje jedra: `sysctl`.
- Datotečni sistem `/proc/`.

```
uname -a  
sysctl -a  
man proc  
cat /proc/version
```

Izvorna koda

- Eden največjih „open-source“ projektov.
 - Več milijonov vrstic kode.
 - GPL licenca.
- Se nahaja v `/usr/src/linux-source-X.Y.Z`
 - Namestitev: `apt-get install linux-source`
- <http://www.kernel.org>
 - <http://www.linux.fm/> :)

Sistemske klici



Vrste sistemskih klicev

- Tabela *rokovalnikov* sistemskih klicev.
- *Indeks rokovalnika* v tabeli je številka *sistemskega klica*.
- Vrsti red je pomemben in se pri posodobitvah jedra ne spreminja.

v3.2.0

```
/* arch/x86/kernel/  
syscall_table_32.S */  
  
ENTRY(sys_call_table)  
    .long sys_restart_syscall  
    .long sys_exit  
    .long sys_fork  
    .long sys_read  
    .long sys_write  
    .long sys_open  
    .long sys_close  
    .long sys_waitpid  
    .long sys_creat  
    .long sys_link  
    .long sys_unlink  
    .long sys_execve  
    .long sys_chdir  
    .long sys_time  
    .long sys_mknod  
    .long sys_chmod  
    ...
```

v4.3

- Tabela rokovalnikov:
 - arch/x86/entry/**syscall_32.c**
 - tabela ia32_sys_call_table[]
 - makro __SYSCALL_I386
- Include <asm/syscalls_32.h>
 - potrebno prevajanje!
 - arch/x86/include/**generated/asm/syscalls_32.h**
- Tabelo sicer najdemo v:
 - arch/x86/entry/**syscalls/syscall_32.tbl**
 - zaženi: bash syscalltbl.sh syscall_32.tbl /dev/stdout

Izvedba sistemska klica

Priprava na vstop v jedro.

- Št. sis. klica: `eax`
- Parametri: `ebx`, `ecx`, `edx`, `esi`, `ebp`

Vstop v jedro.

- Ukaz `int 0x80`, `sysenter` oz. `syscall`.

Rokovalnik pasti sistemskih klicev.

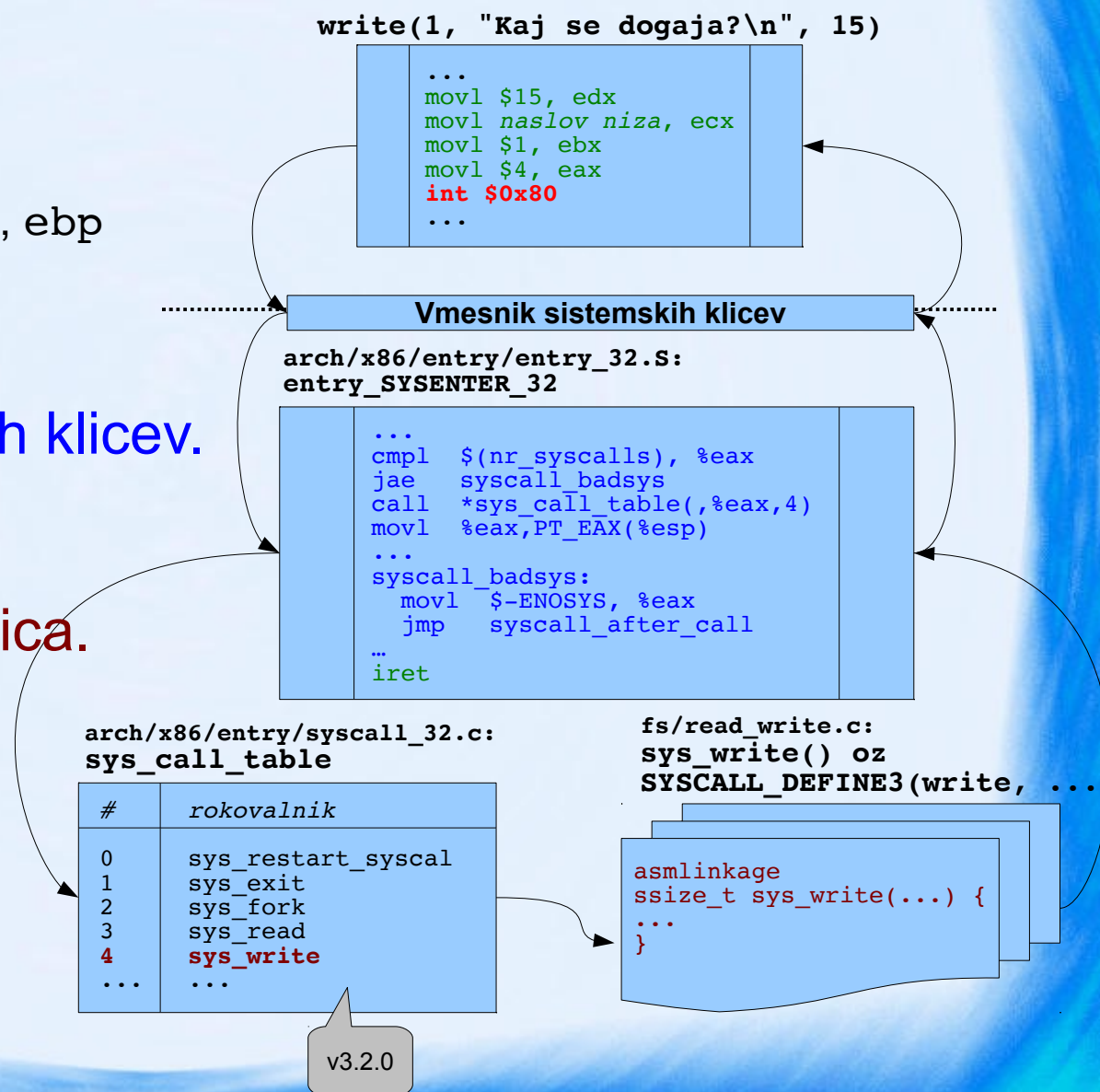
- Obdelava številke sis. klica.
- Poišče rokovalnik v tabeli rokovalnikov.

Rokovalnik sistemskega klica.

- Npr. `sys_write()`.

Izstop iz jedra.

- Ukaz `iret`, `sysexit` oz. `sysret`.
- Register `eax`: izhodni status



Kje kaj najdem?

v3.2.0

```
/* arch/x86/kernel/  
syscall_table_32.S */  
  
ENTRY(sys_call_table)  
    .long sys_restart_syscall  
    .long sys_exit  
    .long sys_fork  
    .long sys_read  
    .long sys_write  
    .long sys_open  
    .long sys_close  
    .long sys_waitpid  
    .long sys_creat  
    .long sys_link  
    .long sys_unlink  
    .long sys_execve  
    .long sys_chdir  
    .long sys_time  
    .long sys_mknod  
    .long sys_chmod  
    ...
```

v3.14

- Tabela rokovalnikov:
 - arch/x86/kernel/syscall_32.c
 - tabela sys_call_table[]
 - makro __SYSCALL_I386
- Include <asm/syscalls_32.h>
 - potrebno prevajanje!
 - arch/x86/include/generated/asm/syscalls_32.h
- Tabelo sicer najdemo v:
 - arch/x86/syscalls/syscall_32.tbl
 - glej tudi skripte, npr. syscallhdr.sh

v4.3

- Tabela rokovalnikov:
 - arch/x86/entry/**syscall_32.c**
 - tabela ia32_sys_call_table[]
 - makro __SYSCALL_I386
- Include <asm/syscalls_32.h>
 - potrebno prevajanje!
 - arch/x86/include/**generated/asm/syscalls_32.h**
- Tabelo sicer najdemo v:
 - arch/x86/entry/**syscalls/syscall_32.tbl**
 - zaženi: bash syscalltbl.sh syscall_32.tbl /dev/stdout

Ovojna funkcija

- Ovojna funkcija (*wrapper function*).
 - Uporabniku prijazna funkcija v sistemski knjižnici.
 - Priprava na sistemski klic in njegova izvedba.
 - Večkrat dvoumno poimenovanje:
 - ovojna funkcija `fork()`;
 - sistemski klic številka 2;
 - rokovalnik sistema klica `sys_fork()`.

glibc

- GNU C Library
- Standardna knjižnica za jezik C.
- Del GNU projekta.
- [*http://www.gnu.org/s/libc/*](http://www.gnu.org/s/libc/)
- Vsebuje ogromno funkcij:
 - med drugim tudi ovojne funkcije sistemskih klicev.

glibc

- Izvedba systemskega klica.
 - Preko ovojne funkcije.
 - `open()`, `close()`, `read()`, `write()`,
 - `fork()`, `exec()`, `wait()`, itd.
 - Preko `syscall()`.
 - `man syscall`
 - `syscall(__NR_write, 1, "Pozdravljen, svet!\n", 19);`
 - Posredno preko ostalih funkcij.
 - `printf()`, itd.

glibc

- Lovljenje napak.
 - Izhodni status funkcije 0 (uspeh) ali -1 (neuspeh).
- Spremenljivka `errno`.
 - Koda napake.
 - `man errno`
- Izpis obvestila o napaki.
 - Funkcija `perror()`.

```
#include <stdlib.h>
#include <errno.h>

int main(int argc, char* argv[]) {
    if (chdir(argv[1]) < 0) {
        int koda = errno;
        perror(argv[0]);
        exit(koda);
    }
    write(1, "Vse je OK.\n", 11);
    exit(0);
}
```

Sistemske klici

- Statistika sistemskih klicev

- Ukaz time.
- Ukaz strace.
- Ukaz ltrace.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>

int main() {
    printf("Kaj se dogaja?\n");
    printf("Uporaba funkcije printf\n");
    printf("Adijo, svet!\n");
    mkdir("imenik", S_IRUSR | S_IWUSR | S_IXUSR);
    printf("PID = %i\n", getpid());
    printf("PPID = %i\n", getppid());
    exit(42);
}
```

```
gcc -static -o prog prog.c
strace ./prog
strace -c ./prog
```


Sistemske klici

- Branje atributov datoteke.
 - `stat(path, buf)`
- Nastavljanje lastnika in skupine.
 - `chown(path, uid, gid)`
- Nastavljanje dovoljenj.
 - `chmod(path, mode)`
- Preverjanje dovoljenj.
 - `access(path, mode)`

Sistemske klici

- Ustvarjanje trde povezave.
 - `link(path, target)`
- Brisanje datoteke.
 - `unlink(path)`
- Preimenovanje datoteke.
 - `rename(src, dest)`
- Simbolične povezave.
 - `symlink(path, target)`
 - `readlink(path, buf, bufsize)`

Sistemske klici

- Trenutni delovni imenik.
 - `getcwd(buffer, size), chdir(path)`
- Ustvarjanje in brisanje imenika.
 - `mkdir(path, mode), rmdir(path)`
- Branje imenika.
 - `opendir(), readdir(), closedir()`