

# Operacijski sistemi



Navidezni  
datotečni sistem

# Vsebina

- Imeniška struktura
- Pripenjanje naprav
- Navidezni datotečni sistem
- Datotečni deskriptorji

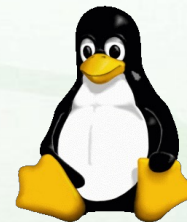
# Imeniška struktura

- Različni pristopi
  - hierarhija imenikov
  - imeniški prostori (Partner 10 prostorov)
- Problem: več medijev
  - Pripenjanje več pomnilnih naprav oz. medijev
    - dostop do datotek v različnih napravah
    - vsaka naprava ima svojo imeniško strukturo
  - Kako omogočiti dostop do različnih naprav?



# Imeniška struktura

- Ločevanje med datotekami različnih naprav
  - **več ločenih** imeniških struktur
    - Windows
  - **ena enotna** imeniška struktura
    - Linux, Unix, macOS



# Imeniška struktura



- **Več imeniških struktur**
  - črkovne oznake naprav
    - disketni enoti: A:, B:, diskovne enote: C:, D:, ...
    - ob pripetju naprave se ji avtomatsko dodeli prosta črka
  - vsaki oznaki ustreza imeniška struktura naprave
    - dostop je preko oznake naprave in poti do datoteke
  - polno ovrednoteno ime
    - vsebuje napravo in absolutno pot
    - npr. C:\Users\Jure\OS-skrivnosti.docx

# Imeniška struktura



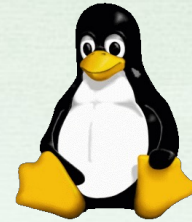
- **Enotna imeniška struktura**

- ena imeniška struktura
- naprave so dostope preko nekaterih imenikov
  - celotno imeniško strukturo dane naprave pripnemo kot pod-strukturo na podani imenik

- **Korenski datotečni sistem**

- osnovna imeniška struktura izbrane naprave
- se pripne ob zagonu OS na korenski imenik
- vsebuje datoteke, pomembne za zagon OS

# Pripenjanje naprav



- Pripenjanje (montiranje, mount)
  - pripenjanje dodatnega datotečnega sistema v obstoječo imeniško strukturo
  - **točka pripenjanja** (mount point)
    - *ciljni imenik* v obstoječi imeniški strukturi, kamor želimo pripeti dodatni datotečni sistem
      - npr. navadno nek prazen imenik, `/mnt/usb`
    - točka pripenjanja se **prekrije** s korenskim imenikom dodatnega datotečnega sistema
      - prvotne datoteke za čas priklopa niso dostopne
  - ukaza: `mount` in `umount`



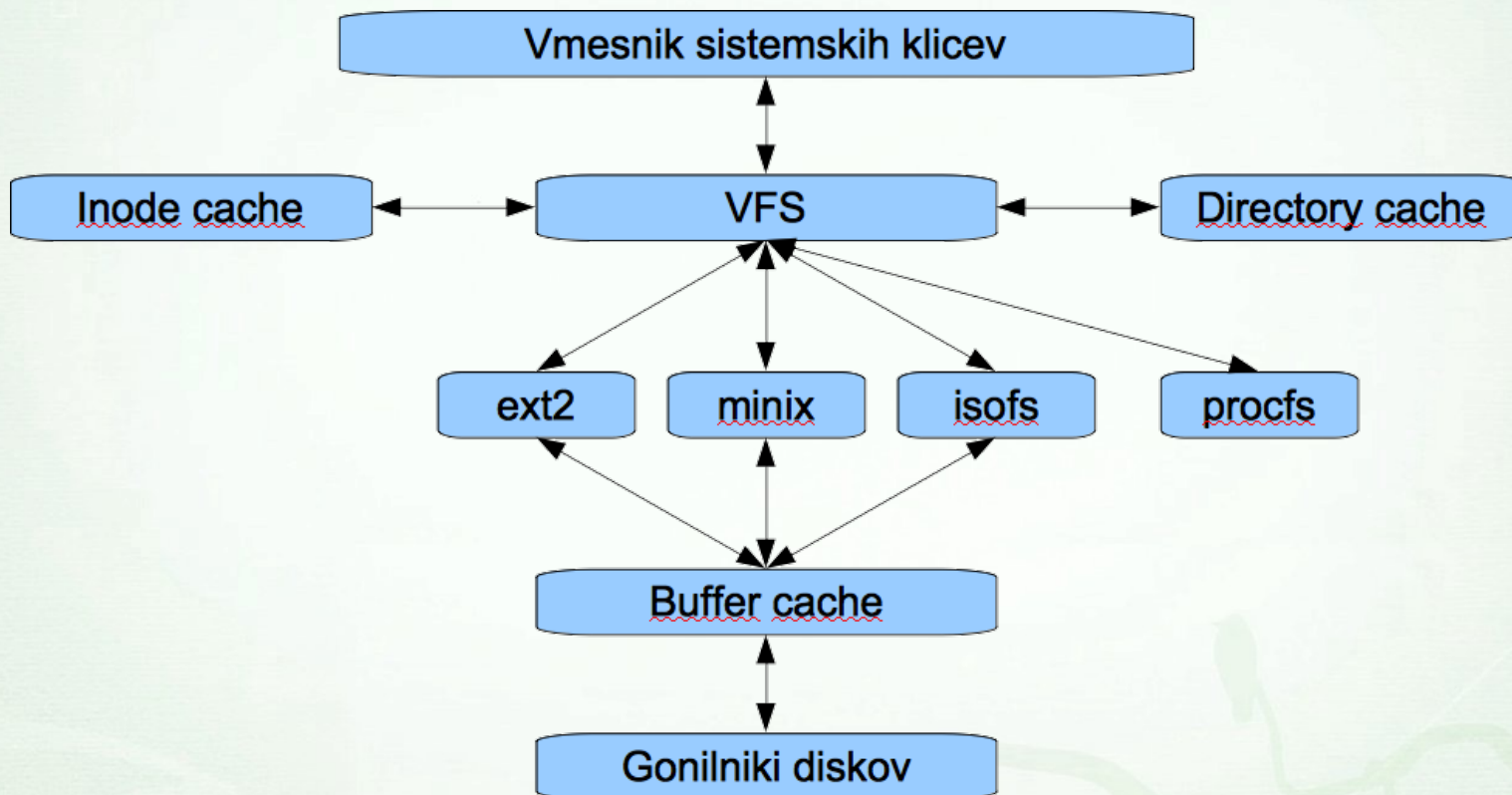
# Navidezni datotečni sistem

- Navidezni datotečni sistem (VFS)
  - različni (fizični) datotečni sistemi podpirajo podobne koncepte
  - uporabniku nudi enoten vmesnik do različnih fizičnih datotečnih sistemov
  - nudi tudi abstrakcijo datoteke
    - datoteka kot vir, različni tipi datotek
  - “standardna” hierarhija imenikov



# Navidezni datotečni sistem

- Struktura VFS



# Navidezni datotečni sistem

- Objektna orientiranost
  - objekti izvedeni kot strukture v C
  - metode kot ustrezne funkcije s parametrom self

```
// Header
struct Point; // forward declared for encapsulation
Point* Point__create(int x, int y); // equivalent to "new Point(x, y)"
void Point__destroy(Point* self); // equivalent to "delete point"
int Point__x(Point* self); // equivalent to "point->x()"
int Point__y(Point* self); // equivalent to "point->y()"
```

- VFS objekti
  - organizacija datotek: **superblock, inode, dentry**
  - datotečni deskriptor: **file**

# Navidezni datotečni sistem

- Struktura **superblock**
  - predstavitev priklopljenega datotečnega sistema

## Lastnosti

- naprava, kjer se nahaja datotečni sistem
- tip datotečnega sistema
- velikost bloka
- zastavice: read-only, dirty, ...
- kazalec inode na **korenski imenik**
- nizko nivojske operacije nad datotečnim sistemom



# Navidezni datotečni sistem

- Struktura **inode** (index node)
  - datoteka poljubnega tipa
  - predstavlja vse razen imena datoteke

## Lastnosti

- št. inode, št. trdih povezav, velikost datoteke
- lastnik, skupina, dovoljenja
- datum/čas dostopa oz. spremembe
- kazalci na bloke z vsebino
- operacije
  - create(), link(), unlink(), symlink(), readlink(), ...
  - mkdir(), rmdir(), rename(), ...
  - read(), write(), open(), close(), ...

# Navidezni datotečni sistem

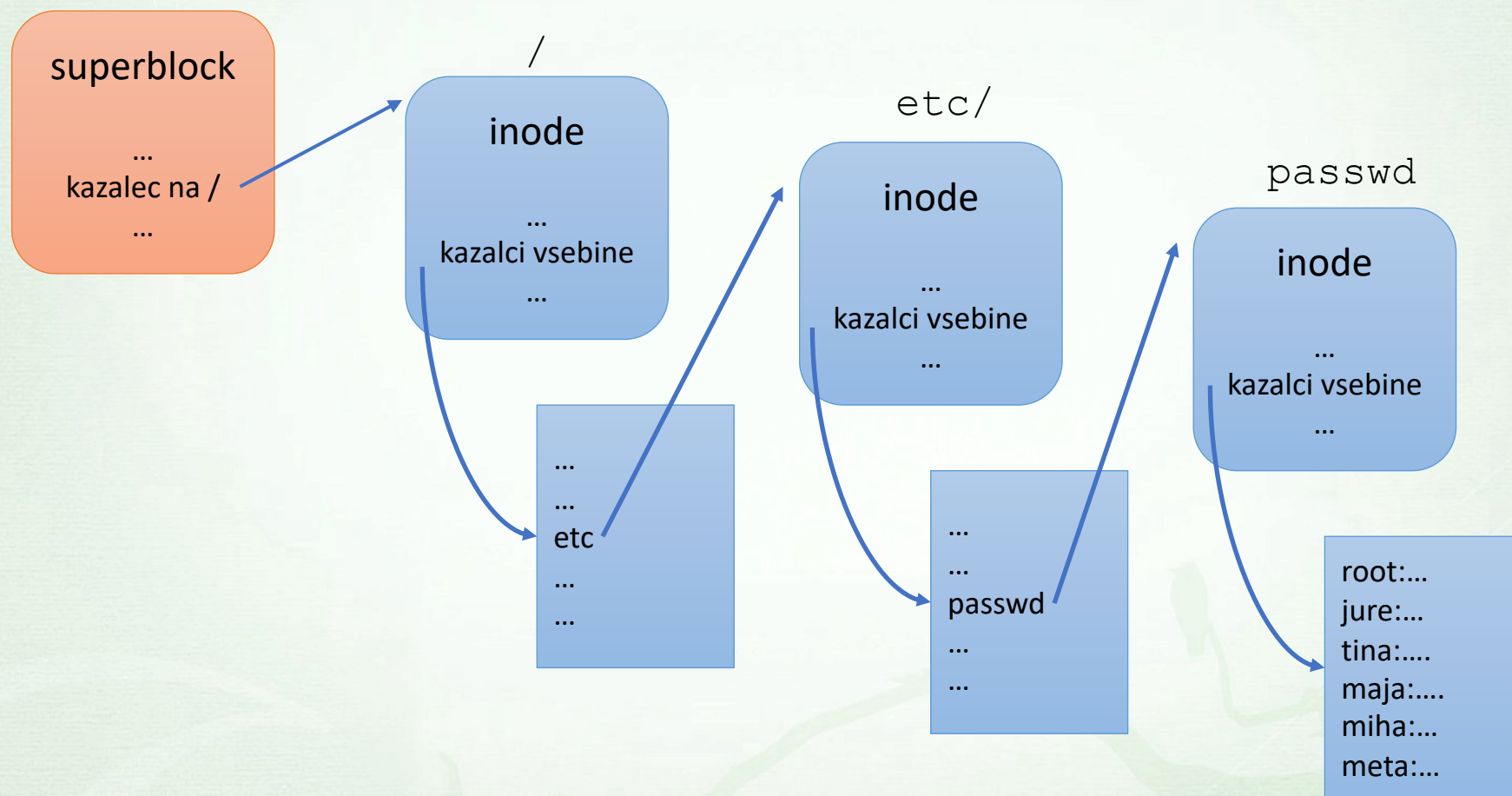
- Struktura **dentry** (directory entry)
  - imenski vnos v imeniku
  - preslikava med *imeni* in *inodei*

## Lastnosti

- ime datoteke
- kazalec inode, ki predstavlja datoteko
- kazalec na starševski imenik
- števci uporabe

# Navidezni datotečni sistem

- Predstavitev `/etc/passwd`





# Datotečni deskriptorji

- Struktura **file**

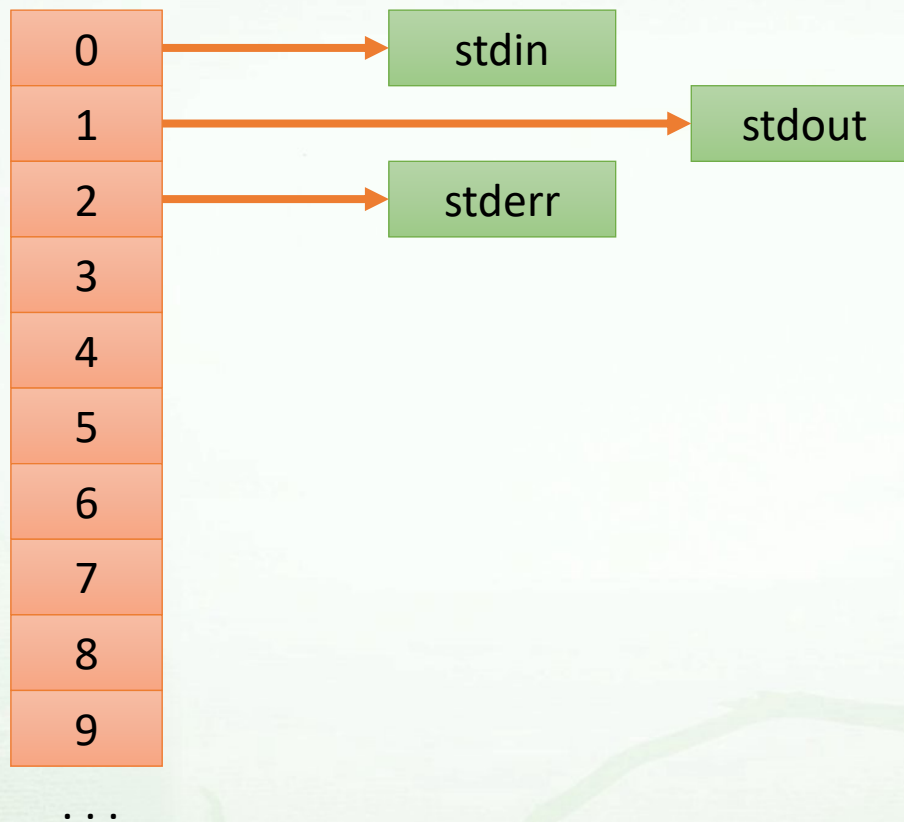
- predstavlja odprto datoteko nekega procesa
- datotečni deskriptor
  - vsaki št. datotečnega deskriptorja ustreza ta objekt

## Lastnosti

- kazalec na ustrezen **dentry**
- datotečni sistem
- števec uporabe
- uid in gid procesa, ki je odprl datoteko
- **pozicija** v datoteki

# Datotečni deskriptorji

- Številka datotečnega deskriptorja



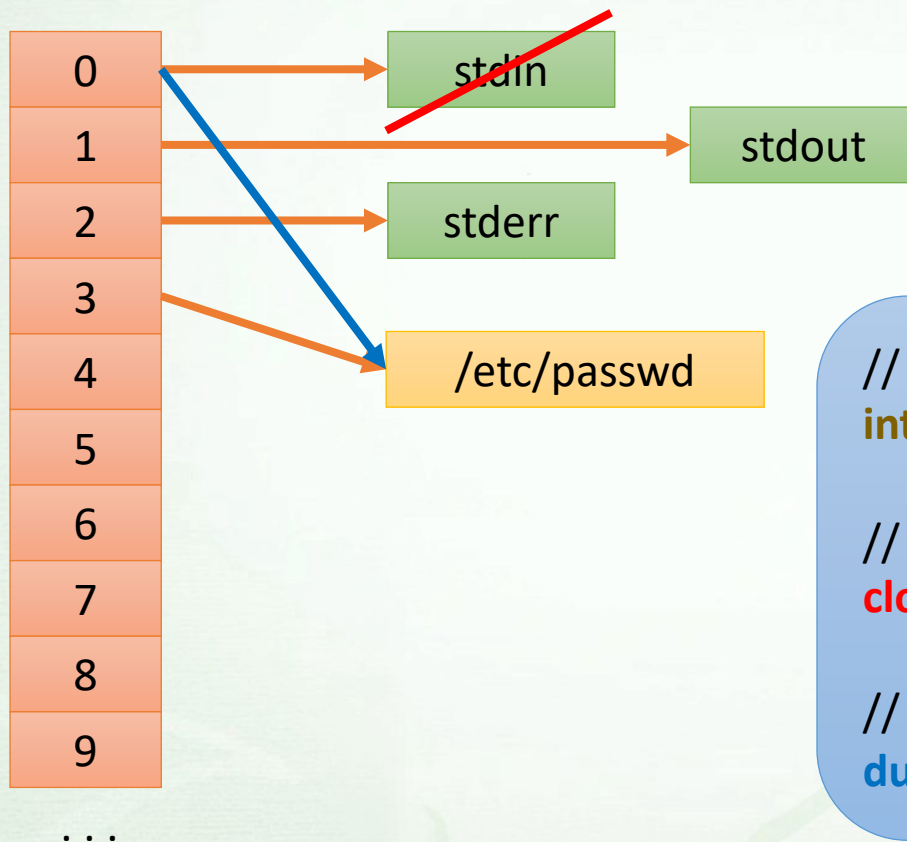
# Datotečni deskriptorji

- Podvajanje št. deskriptorja
  - **sistemski klic `dup(fd)`**
    - *new* = `dup(orig)`
    - vrne št. deskriptorja *new*, ki predstavlja isti deskriptor oz. odprto datoteko kot *orig*
- lastnosti
  - uporabi se prva prosta številka deskriptorja
  - nova in stara številka predstavljata isti deskriptor (isti objekt *file*)
  - duplikati si delijo pozicijo v datoteki ipd.



# Datotečni deskriptorji

- Podvajanje št. deskriptorja
  - primer uporabe



```
// odpremo datoteko  
int fd = open("/etc/passwd", O_RDONLY);
```

```
// zapremo obstoječi deskriptor  
close(0);
```

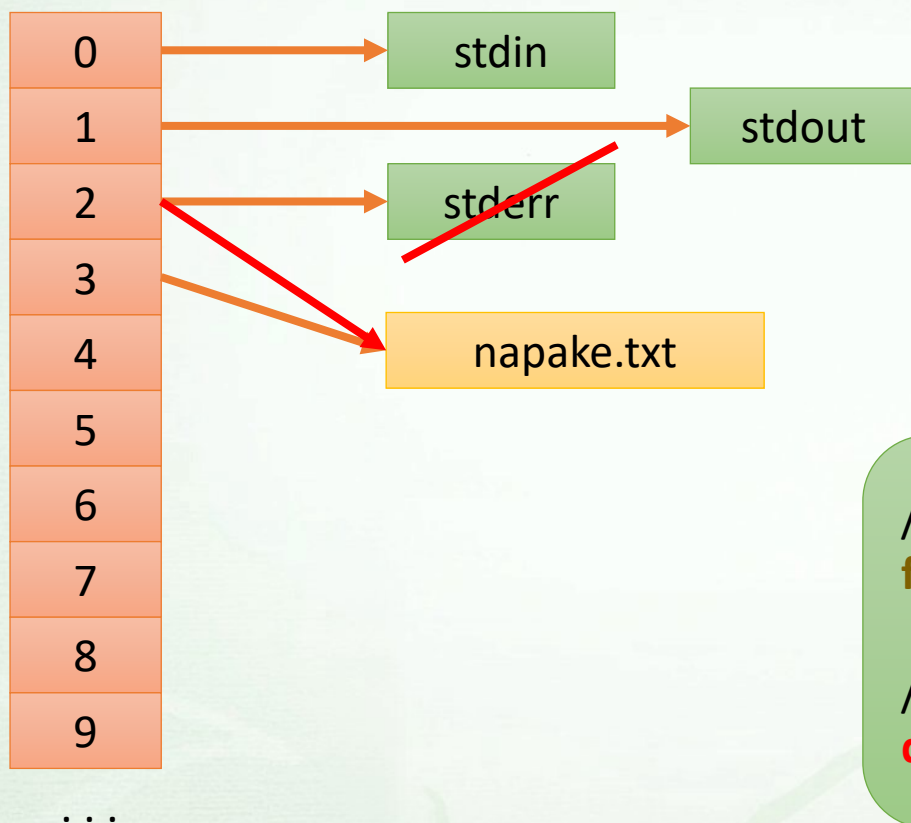
```
// podvojimo deskriptor  
dup(fd);
```

# Datotečni deskriptorji

- Podvajanje št. deskriptorja
  - **sistemski klic `dup2(orig, new)`**
    - preusmeritev deskriptorja *orig* v *new*
    - če je *new* že v uporabi, se prej zapre

# Datotečni deskriptorji

- Podvajanje št. deskriptorja
  - primer uporabe



```
// odpremo datoteko  
fd = open("napake.txt", O_WRONLY);
```

```
// izvedemo preusmeritev  
dup2(fd, 2);
```