

Sistemska programiranje

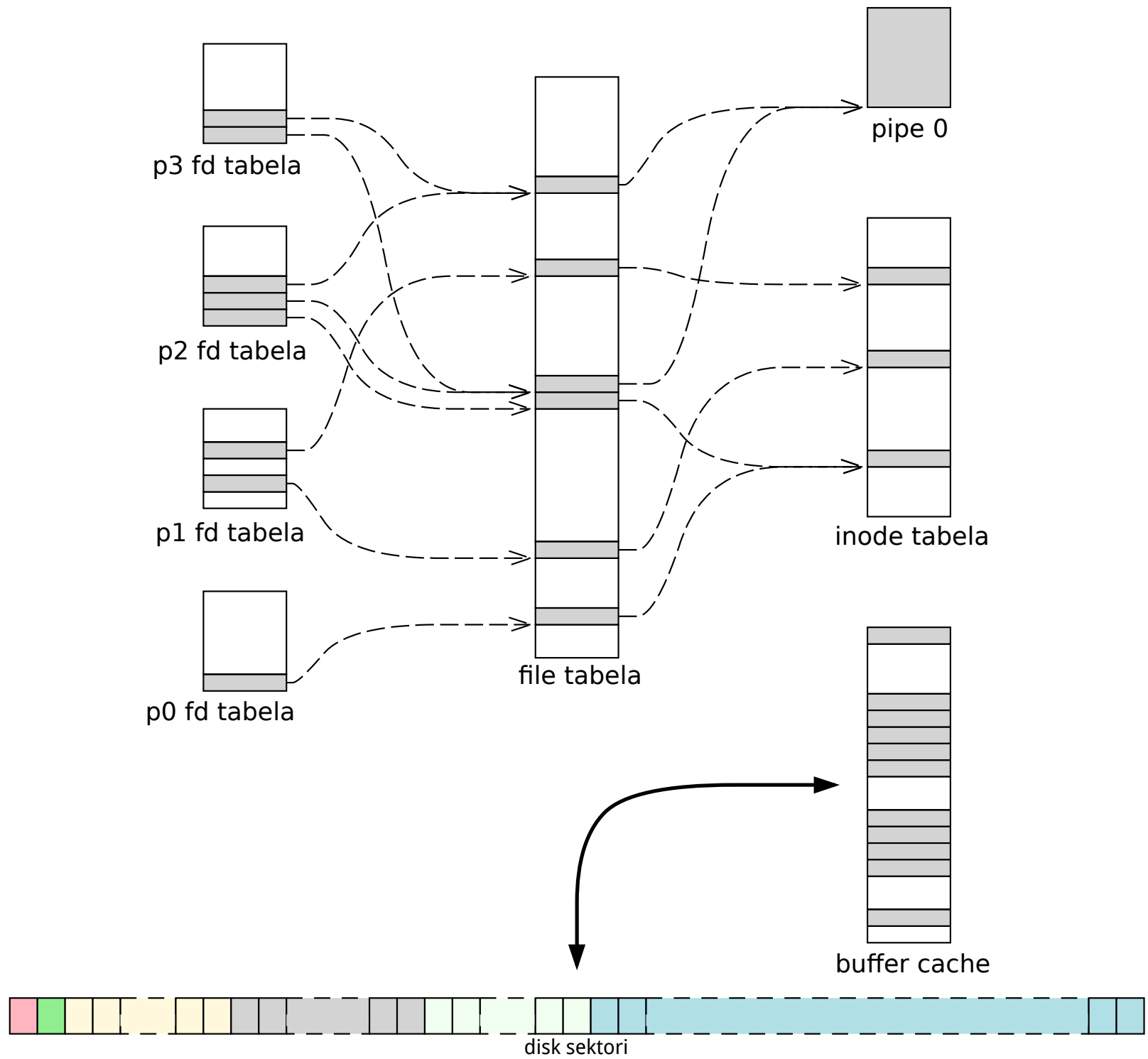
dr.sc. Amer Hasanović

Fajl deskriptor

- Koncept putem kojeg kernel uspostavlja vezu između disk inode-a i određenog procesa
- Svaki proces ima tabelu koja mapira cijele brojeve u otvorene fajlove. Broj u mapiranju predstavlja fajl deskriptor.
- Fajl deskriptori 0, 1 i 2 su inicijalno u svakom procesu asocirani sa terminal uređajem, i imaju slijedeće uloge:
 - 0 → standardni ulaz (stdin)
 - 1 → standardni izlaz (stdout)
 - 2 → izlaz za greške (stderr)
- Veza između deskriptora i disk inode-a nije direktna i uključuje više objekata koje kernel alocira prilikom formiranja relacije.

Fajl sistem kernel RAM elementi

- Svaki deskriptor je u kernelu asociran sa tačno jednim objektom tipa fajl.
- Fajl je asociran sa inode-om koji predstavlja RAM kopiju disk inode-a.
 - ima ofset, lokaciju sa koje će se vršiti slijedeće čitanje ili pisanje sadržaja iz asociranog inode-a.
- Svaki inode-e je povezan sa proizvoljnim brojem bafera, RAM blokova koji predstavljaju keširane kopije sektora sa diska u kojim se čuva sadržaj disk inode-a asociranim sa RAM inode-om.
- Kernel omogućava da više procesa koristi isti fajl i/ili inode RAM objekat.
 - Sistemskim pozivom fork, fajl deskriptori roditelja kopiraju se u dijete proces.



Tretman iznimki

- Unix API je baziran na programskom jeziku C
 - Ozbiorom da C ne podržava iznimke, signalizacija iznimnih stanja unutar sistemskih poziva se realizira putem error kodova.
 - Pozivi rezerviraju posebne povratne vrijednosti funkcija kojim se signalizira iznimno stanje:
 - najčešće -1, 0 ili NULL
 - globalna varijabla errno, definirana unutar errno.h, dodatno opisuje iznimno stanje.
 - funkcija perror na stderr izlazu ispisuje tekst koji opisuje iznimno stanje
 - funkcija strerror na osnovu proslijeđeno

Fajl sistem API - open

- Sve FS operacije sa stanovišta procesa operiraju na deskriptorima.
- Prilikom poziva open vrši se autorizacija spram efektivnih ID vrijednosti procesa koji obavlja poziv.
- `int open(const char *pathname, int flags, mode_t mode);`
 - formira konekciju između disk inode-a na datoj pathname stazi sa najmanjim fajl deskriptorom koji trenutno nije u upotrebi.
 - bitni flagovi:
 - `O_RDONLY`, `O_WRONLY`, `O_RDWR` → čitanje, pisanje, čitanje i pisanje.
 - `O_APPEND` → svako pisanje događa se na kraju fajla.
 - `O_TRUNC` → ukoliko fajl postoji i otvoren je u modu za pisanje postavlja dužinu fajla na 0.
 - `O_SYNC` – Svako pisanje forsira IO operaciju na uređaju.

- O_CREAT → kreira fajl ako ne postoji.
- O_EXCL – Signalizira grešku ukoliko se koristi O_CREAT a fajl već postoji.

```
int fd = open(path, O_WRONLY | O_CREAT | O_EXCL, 0777);  
if ((fd == -1) && (EEXIST == errno)) {  
    fd = open(path, O_WRONLY);  
    // ...  
}
```

Fajl sistem API - close

- `int close(int fd);`
- Otpušta kernel resurse asocirane sa fd deskriptorom.
- Potrebno pozivati prije terminiranja opsega u kojem se koristi deskriptor
- Kernel automatski poziva close na svim deskriptorima prije terminiranja procesa.
- Nakon poziva close, broj fd će biti slobodan za korištenje prilikom slijedećeg poziva open.

Fajl sistem API - read

- `ssize_t read(int fd, void *buff, size_t nbytes);`
- Upisuje `nbytes` bajta pročitanih iz fajla asociranim sa deskriptorom `fd` u memorijsku lokaciju datu pointerom `buff`.
- Vraća broj prenešenih bajta koji može biti manji od `nbytes`.
- Čitanje se vrši sa lokacije trenutnog offseta fajla koji je asociran sa deskriptorom. Offset se inkrementira za broj prenešenih bajta.

Fajl sistem API - write

- `ssize_t write(int fd, void *buff, size_t nbytes);`
- Upisuje `nbytes` bajta pročitanih sa memorijske lokacije `buff` u fajl asociran sa deskriptorom `fd`.
- Vraća broj prenešenih bajta koji može biti manji od `nbytes`.
- Pisanje se vrši na lokaciju trenutnog offseta fajla koji je asociran sa deskriptorom. Offset se inkrementira za broj prenešenih bajta.
 - Ukoliko je fajl otvoren sa `O_APPEND` pisanje se uvijek vrši na kraju fajla.

lseek, fcntl, fstat, dup i dup2

- lseek
 - Mijenja trenutnu poziciju ofseta u fajlu asociranim sa određenim deskriptorom
- fcntl
 - Postavlja ili čita vrijednosti postavki fajla asociranog sa određenim deskriptorom.
- fstat
 - Vraća informacije o fajlu asociranim sa deskriptorom, ili sa stazom ako se koristi kao stat.
- dup ili dup2
 - Prave kopiju deskriptora.

```

#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define BUFFSIZE 512

int main(void) {
    int n;
    char buf[BUFFSIZE];
    int flags;
    if ((flags = fcntl(STDOUT_FILENO, F_GETFL, 0)) < 0) {
        perror("Can't get file descriptor flags");
        exit(1);
    }
    flags |= O_SYNC;
    if (fcntl(STDOUT_FILENO, F_SETFL, flags) < 0) {
        perror("Can't set file descriptor flags");
        exit(1);
    }

    while ((n = read(STDIN_FILENO, buf, BUFFSIZE)) > 0 )
        if ( write(STDOUT_FILENO, buf, n) != n ) {
            fprintf(stderr, "write error\n");
            exit(1);
        }

    if (n < 0) {
        fprintf(stderr, "read error\n");
        exit(1);
    }
    return 0;
}

```