



Operativni sistemi

# Sistemsko programiranje

## Međuprocesna komunikacija I

Katedra za računarstvo  
Elektronski fakultet u Nišu

Prof. dr Dragan Stojanović

Prof. dr Aleksandar Stanimirović

Prof. dr Bratislav Predić



# Sadržaj

- UNIX/Linux međuprocesna komunikacija
- Datavodi
- Preusmeravanje U/I operacija
- Imenovani datavod



# Sadržaj

- UNIX/Linux međuprocesna komunikacija
- Datavodi
- Preusmeravanje U/I operacija
- Imenovani datavod



# UNIX/Linux međuprocesna komunikacija

## Pojam

- Procesi imaju međusobno **nezavisne adresne prostore**.
- Svaka komunikacija i razmena podataka između različitih procesa zahteva korišćenje **posebnih mehanizama za komunikaciju**.
- UNIX/Linux operativni sistemi podržavaju različite mehanizme za komunikaciju između procesa.
- Najzastupljeniji mehanizmi za komunikaciju između procesa su:
  - ▶ **datavodi**
  - ▶ **signali**
  - ▶ **redovi poruka**
  - ▶ **deljiva memorija**



# Sadržaj

- UNIX/Linux međuprocesna komunikacija
- Datavodi
- Preusmeravanje U/I operacija
- Imenovani datavodi



# Datavodi

## Pojam

- **Datavod (pipe)** je mehanizam za komunikaciju između procesa koji omogućava **jednosmernu komunikaciju** između dva **istovremeno aktivna procesa** korišćenjem toka (stream) podataka.
- Osnovne karakteristike datavoda su:
  - datavod funkcioniše kao **FIFO red** (prvi upisani podatak na jednom kraju biće prvi pročitani podatak na drugom kraju)
  - operacije čitanja i pisanja nad datavodom su **blokirajuće** pa nema potrebe za posebnim mehanizmima za sinhronizaciju
  - nije dozvoljeno upisivati podatke u datavod ako ne postoji proces koji čita te podatke
  - ukoliko ne postoji proces koji upisuje podatke u datavod, datavod simulira **EOF (end of file)**



# Datavodi

## Kreiranje datavoda

```
#include <unistd.h>  
int pipe (int p[2]);
```

## Semantika

- Sistemski poziv koji **kreira novi datavod** i **smešta par deskriptora koji ukazuju na krajeve datavoda** u niz p:
  - p[0] – **kraj datavoda koji se koristi za čitanje**
  - p[1] – **kraj datavoda koji se koristi za pisanje**
- Funkcija vraća 0 u slučaju uspešnog izvršenja odnosno -1 ukoliko je došlo do greške.



# Datavodi

## Zatvaranje datavoda

```
#include <unistd.h>  
int close (int fd);
```

## Semantika

- Sistemski poziv koji **zatvara deskriptor** kojim se pristupa jednom od krajeva datavoda.
- Kada se u **svim procesima zatvore svi deskriptori** koji ukazuju na datavod datavod se automatski uništava.





# Datavodi

## Čitanje i pisanje iz datavoda

```
#include <unistd.h>
ssize_t read(int fd, void * buff, size_t count);
ssize_t write(int fd, void * buff, size_t count);
```

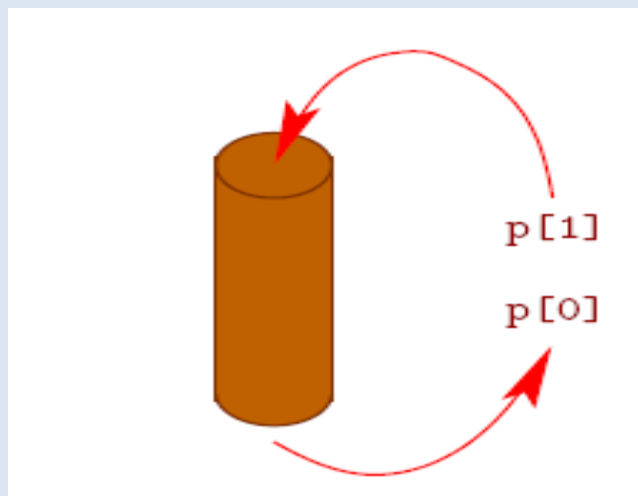
## Semantika

- Sistemski pozivi koji se koriste za **čitanje** odnosno **pisanje podataka** iz datavoda.
- Prvi argument predstavlja **deskriptor odgovarajućeg kraja datavoda**.
- Drugi argument predstavlja pokazivač na bafer iz koga se podaci upisuju u datavod ili u koji se podaci upisuju iz datavoda.
- Treći argument predstavlja veličinu bafera u bajtovima.
- Povratna vrednost funkcija predstavlja broj pročitanih bajtova odnosno broj upisanih bajtova u datavod.

# Datavodi

## Komunikacija korišćenjem datavoda

- Da bi se datavod mogao koristiti za komunikaciju između dva procesa treba obezbediti da **deskriptori budu vidljivi i validni u oba procesa**.
- Da bi dva procesa mogla da komuniciraju korišćenjem datavoda potrebno je da budu u srodstvu (roditelje dete ili da imaju zajedničkog pretka).
- Dva postojeća procesa je nemoguće povezati datavodom.





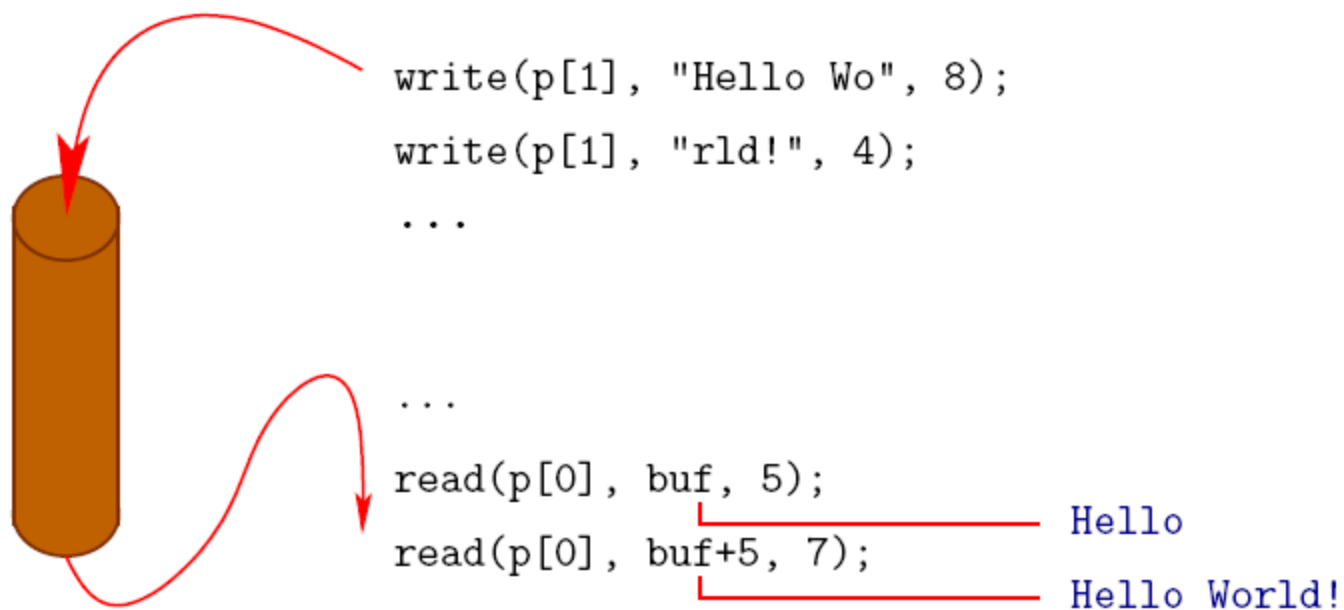
# Datavodi

## Komunikacija korišćenjem datavoda

- Podaci upisani u datavod ostaju u datavodu sve dok ih drugi proces ne pročita. Nakon toga se zamenjuju novoupisanim podacima.
- U trenutku kada se **datavod napuni** (zapisan je maksimalan broj bajtova), sistem **blokira operaciju upisivanja** sve dok se memorija datavoda ne oslobodi operacijom za čitanje podataka.
- Ako je **datavod prazan** operacija čitanja podataka blokira proces sve dok drugi proces u datavod ne upiše neke podatke ili se datavod ne deaktivira.
- Upisivanje i čitanje podataka se vrši po FIFO principu.
- Podaci koji se nalaze u datavodu **nisu struktuirani** odnosno tretiraju se kao niz bajtova.

# Datavodi

## Komunikacija korišćenjem datavoda



# Datavodi

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
```

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

```
{
    int pd[2];
    int pid;
    char buff[255];
```

Kreiranje datavoda.

```
if ( pipe(pd) < 0 )
    exit(1);
```

```
pid = fork();
if ( pid == 0 )
{
    close( pd[1] );
    read( pd[0], buff, 255 );
    printf( "%s\n", buff );
    close( pd[0] );
}
```

Dete čita podatke iz datavoda.

```
else
```

```
{
    close( pd[0] );
    scanf( "%s", buff );
    write( pd[1], buff, strlen(buff) + 1 );
    wait(NULL);
    close( pd[1] );
}
```

Roditelj upisuje podatke u datavod.



# Datavodi

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
int main (int argc, char *argv[])
{
    int pd[2];
    int pid;
    char buff[255];

    if ( pipe(pd) < 0 )
        exit(1);

    pid = fork();
    if ( pid == 0 )
    {
        close( pd[1] );
        sprintf(buff, "%d", pd[0]);
        execl("proces2", "proces2", buff, NULL);
    }
    else
    {
        close( pd[0] );
        scanf( "%s", buff );
        write( pd[1], buff, strlen(buff)+1);
        wait(NULL);
        close( pd[1] );
    }
}
```

Kraj za čitanje datavoda se prosleđuje kao ulazni argument novog procesa.

Proces dete učitava novu izvršnu datoteku.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main (int argc, char *argv[])
{
    int pd;
    char buff[255];
    if ( argc < 2 )
        exit(1);

    pd = atoi(argv[1]);

    read( pd, buff, 255 );
    printf("%s\n", buff);

    close(pd);
}
```

Deskriptor je prosleđen kao ulazni argument.



# Sadržaj

- UNIX/Linux međuprocena komunikacija
- Datavodi
- Preusmeravanje U/I operacija
- Imenovani datavod



# Preusmeravanje U/I operacija

## Preusmeravanje U/I operacija

```
#include <unistd.h>
int dup(int oldfd);
int dup2(int oldfd, int newfd);
```

## Semantika

- Sistemski poziv **dup** nalazi najmanji slobodan deskriptor datoteke i postavlja ga da ukazuje na istu datoteku na koju ukazuje i deskriptor **oldfd**.
- Sistemski poziv **dup2** zatvara deskriptor **newfd**, ukoliko je otvoren, i modifikuje ga takod a ukazuje na istu datoteku na koju ukazuje i deskriptor **oldfd**.
- Oba sistemska poziva u slučaju uspeha vraćaju kopiju deskriptora **oldfd**, a u slučaju greške vraćaju -1.



# Preusmeravanje U/I operacija

## Implementacija ls | more

```
if ( pipe(pd) < 0 )  
    exit(1);
```

```
pid = fork();  
if ( pid == 0 )  
{  
    close(1);  
    dup(pd[1]);  
    execlp("ls", "ls", NULL);  
}
```

Standardni izlaz se preusmerava na kraj datavoda koji se koristi za pisanje.

```
else  
{  
    close(0);  
    dup(pd[0]);  
    execlp("more", "more", NULL);  
}
```

Standardni ulaz se preusmerava na kraj datavoda koji se koristi za čitanje.



# Preusmeravanje U/I operacija

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

int main()
{
    int fd ;    /* Deskriptor datoteke */

    /* Otvara datoteku sa imenom prosleđenim u prvom argumentu poziva programa */
    fd = open (argv[1], O_CREAT | O_TRUNC | O_RDWR, 0777) ;

    dup2 (fd, 1) ; /* Redirektuje standardni izlaz na prethodno otvorenu datoteku */
    close (fd) ;

    /* Izvršava program čije je ime prosleđeno kao drugi argument poziva programa */
    execvp (argv[2], &argv[2]) ;

    printf ("Kraj\n") ;    /* Ne bi trebalo nikada da se izvrši */
}
```



# Sadržaj

- UNIX/Linux međuprocesna komunikacija
- Datavodi
- Preusmeravanje U/I operacija
- Imenovani datavod



# Imenovani datavod

## Kreiranje imenovanog datavoda

```
#include <unistd.h>
int mkfifo(const char * pathname, mode_t mode);
```

## Semantika

- Imenovani datavodi se mogu koristiti za komunikaciju bilo koja dva procesa u sistemu (ne postoji ograničenje vezano za srodstvo procesa).
- Imenovani datavodi su sastavni deo datotečnog sistema.
- Za rad sa imenovanim datavodima mogu se koristiti sve funkcije koje se koriste za rad sa standardnim datotekama.