

# Operacijski sistemi

Medprocesna  
komunikacija

# Vsebina

- Medprocesna komunikacija
  - načini komuniciranja, sporočilo, kanal
- Prenašanje sporočil
  - pošlji in prejmi
  - sinhronost, medpomnenje, naslavljanje
  - učinkovitost in enostavnost uporabe
- Deljeni pomnilnik
  - uporaba, odstranjevanje
- Primerjava

# Medprocesna komunikacija

- **Kaj je medprocesna komunikacija?**
  - komunikacija med procesi
    - navadno gre za sočasne procese
    - prenos podatkov med naslovnimi prostori procesov
  - nadzorovan mehanizem OS, ki omogoča komunikacijo brez kršenja zaščite procesov
- **Namen**
  - izmenjava podatkov
  - sinhronizacija skupnih akcij
  - pohitritev izvajanja (skupaj s sočasnostjo)
  - modularnost (vsak proces za svoje opravilo)

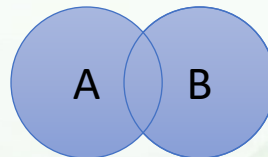


# Medprocesna komunikacija

- Osnovna načina **medprocesne komunikacije**
  - **prenašanje sporočil** (message passing)
    - vtičnice, cevi, sporočilne vrste itd.




- **deljeni pomnilnik** (shared memory)
  - deljeni pomnilnik, pomnilniško preslikane datoteke



# Medprocesna komunikacija

- Ostali načini **medprocesne komunikacije**?
  - argumenti programa
  - okoljske spremeljivke
  - izhodni status (končanega procesa)
  - standardni vhod, izhod in izhod za napake
  - datoteke
  - klic oddaljene procedure
  - sinhronizacija procesov
  - ipd.



Te mehanizme  
obravnavamo  
drugje.

Višjenivojska  
semantika.

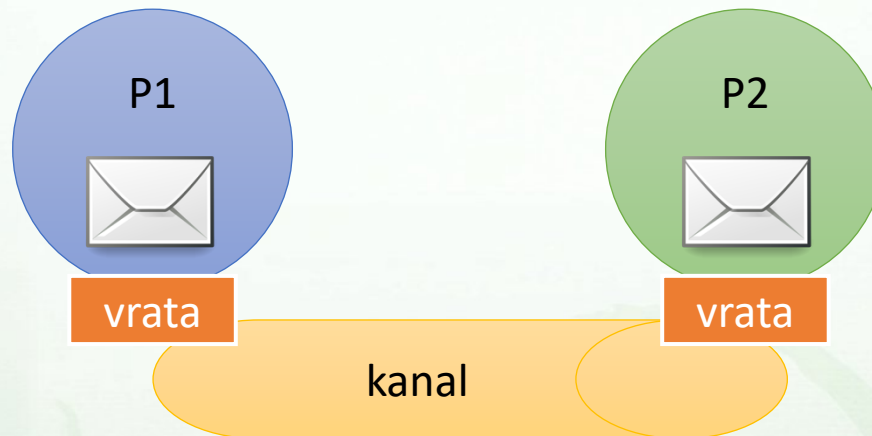
# Medprocesna komunikacija

- **Sporočilo (message)**

- z vidika OS nima pomena
  - le zaporedje bajtov (velikost pogosto omejena)
- z vidika komunicirajočih procesov ima pomen

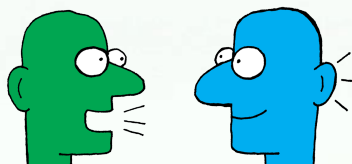
- **Kanal (channel)**

- medij za komunikacijo
  - OS ga ustvari in vzdržuje
  - npr. medpomnilnik, vrsta
- procesi se povežejo na kanal preko *vrat* (port)

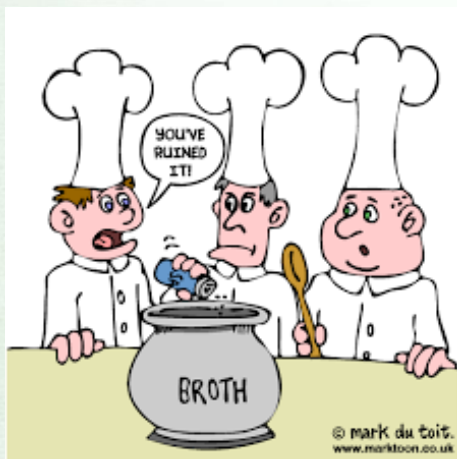




# Medprocesna komunikacija



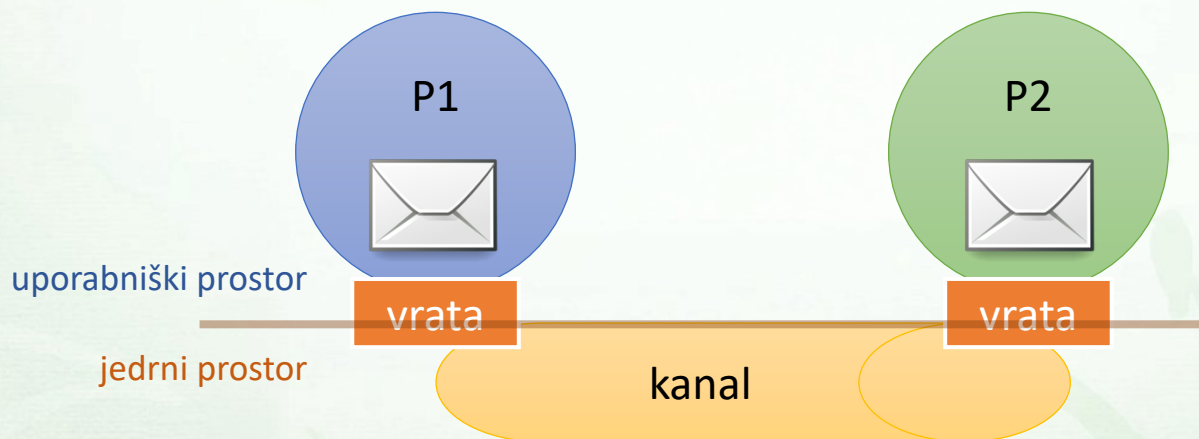
Prenašanje sporočil



Deljeni pomnilnik

# Prenašanje sporočil

- Prenašanje sporočil
  - OS vzpostavi in sprosti kanal
  - OS upravlja kanal
  - OS poskrbi za uspešen prenos in prejem
  - kanal je (navadno) v jedru





# Prenašanje sporočil

- Osnovne operacije oz. programski vmesnik
  - **pošiljanje** sporočila (send, write)
    - pošlje sporočilo v kanal
  - **prejem** sporočila (receive, read)
    - izvede prejem sporočila iz kanala
  - včasih tudi **odgovor** na sporočilo (reply)
    - včasih se na poslano sporočilo zahteva potrditev oz. odgovor

# Prenašanje sporočil

- Sinhronost komunikacije

- **sinhrona operacija**

- operacije blokirajo

- pošiljanje: blokira, dokler sporočilo ni oddano v kanal

- prejem: blokira, dokler ne prejme sporočila

- **zmenek/rendezvous**: kadar sta obe pošlji in prejmi sinhroni operaciji

- **asinhrona operacija**

- operacije ne blokirajo

- pošiljanje: pošlje sporočilo in proces takoj nadaljuje

- prejem: se nadaljuje, četudi sporočila še ni

- potrebujemo medpomnenje sporočil

- npr. vrsta sporočil

# Prenašanje sporočil

- Medpomnenje
  - podatkovna struktura za hranjenje poslanih in še ne prejetih sporočil (vrsta sporočil)
  - **neomejena kapaciteta**
    - pošiljatelj nikoli ne blokira
  - **omejena kapaciteta**
    - pošiljatelj pri polni vrsti blokira, sicer ne
  - **ničelna kapaciteta**
    - pošiljatelj vedno blokira
    - dokler prejemnik ne prejme sporočila



# Prenašanje sporočil

- Naslavljanje procesov

- **neposredna komunikacija**

- procesi se morajo **eksplicitno** poznati
    - naslavljanje preko PID (ga je potrebno poznati)
    - pošlji: kanal se avtomatsko vzpostavi

```
send(pid, msg)  
receive(pid, msg)
```

- **posredna komunikacija**

- procesi komunicirajo preko **vmesnega objekta**
      - npr. nabiralnik, vrata, itd.
    - komunicira lahko več procesov
    - kanal je potrebno predhodno vzpostaviti

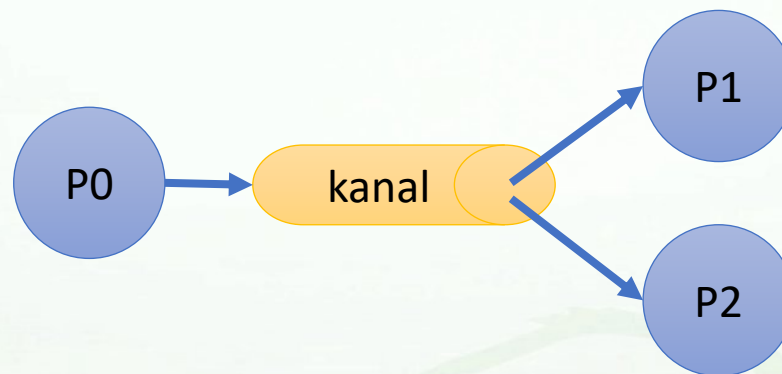
```
send(addr, msg)  
receive(addr, msg)
```

# Prenašanje sporočil

- Posredna komunikacija
  - lokacija vmesnega objekta (nabiralnika)
  - **v naslovnem prostoru procesa**
    - nabiralnik živi le dokler živi proces
    - nabiralnik se sprosti, ko se proces konča
  - **v sistemskem prostoru oz. izven procesa**
    - nabiralnik preživi proces
    - sporočila se nabirajo v kanalu in so na voljo tudi za kasnejše branje
    - OS po potrebi skrbi tudi za čiščenje nabiralnikov

# Prenašanje sporočil

- Posredna komunikacija
  - **več pisalcev**
    - več procesov piše v isti nabiralnik
  - **več bralcev**
    - več procesov bere iz istega nabiralnika
    - Kdo bo prebral sporočilo?

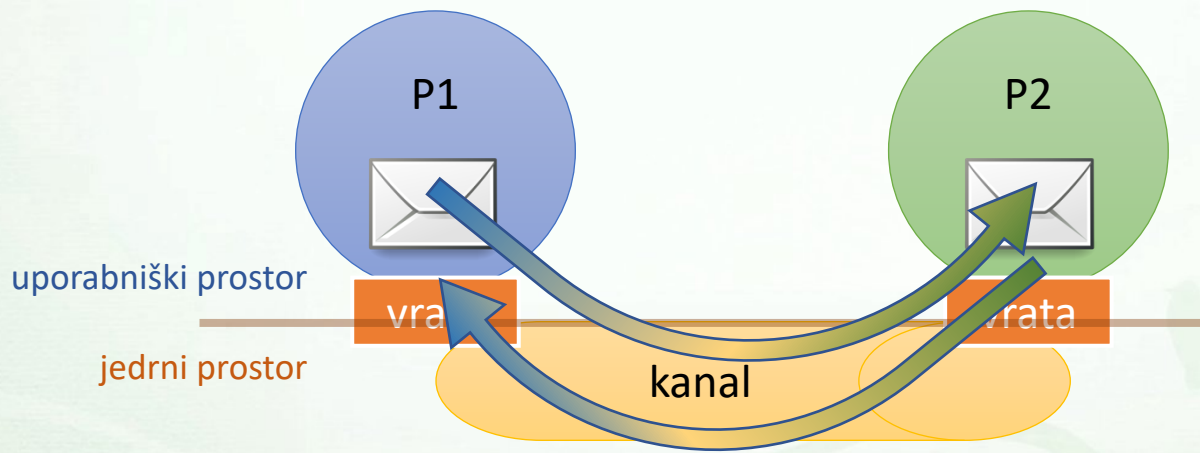




# Prenašanje sporočil

- Učinkovitost

- komuniciranje zahteva sistemske klice
- zahteva-odgovor (request-reply)
  - 4x prehod med uporabniškim in jedrnim načinom
  - 4x kopiranje podatkov

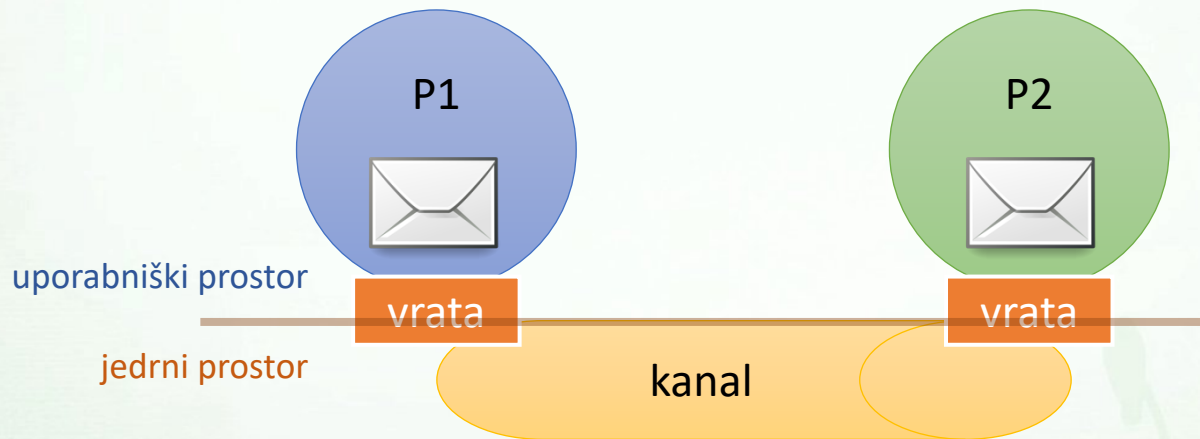


# Prenašanje sporočil

- Učinkovitost – **leno kopiranje**
  - kopiramo le naslov sporočila
  - kopiranje samega sporočila izvedemo kasneje
    - ko prejemnik sproži prejem sporočila
    - sporočilo se kopira neposredno iz pošiljateljevega naslovnega prostora v prejemnikov prostor
    - pošiljatelj se sporočila do prejema ne sme dotikati
  - zahteva-odgovor
    - 4x prehod med uporabniškim in jedrnim načinom
    - 2x kopiranje podatkov

# Prenašanje sporočil

- Enostavnost uporabe
  - OS skrbi za stvaritev in upravljanje kanala
  - OS skrbi za medpomnenje in sinhronizacijo
    - da se podatki ne pokvarijo, prepíšejo itd.





# Prenašanje sporočil

producer-consumer

- Problem **proizvajalec-porabnik**
  - dana sta dva ali več sočasnih procesov
  - proizvajalec P:
    - proizvaja podatke in jih pošilja porabniku
  - porabnik Q:
    - porablja podatke
  - izzivi sočasnosti procesov
    - ni znano, kdo ima procesor
    - Kdaj naj P proizvede nov podatek?
    - Ali Q lahko porabi podatek?

# Prenašanje sporočil

- Problem **proizvajalec-porabnik**

Proizvajalec P

while true do

...

m = produce()

**send**(Q, m)

...

done

Porabnik Q

while true do

...

**receive**(P, m)

consume(m)

...

done

- Kako velika je vrsta sporočil med P in Q?
- So operacije sinhronne?

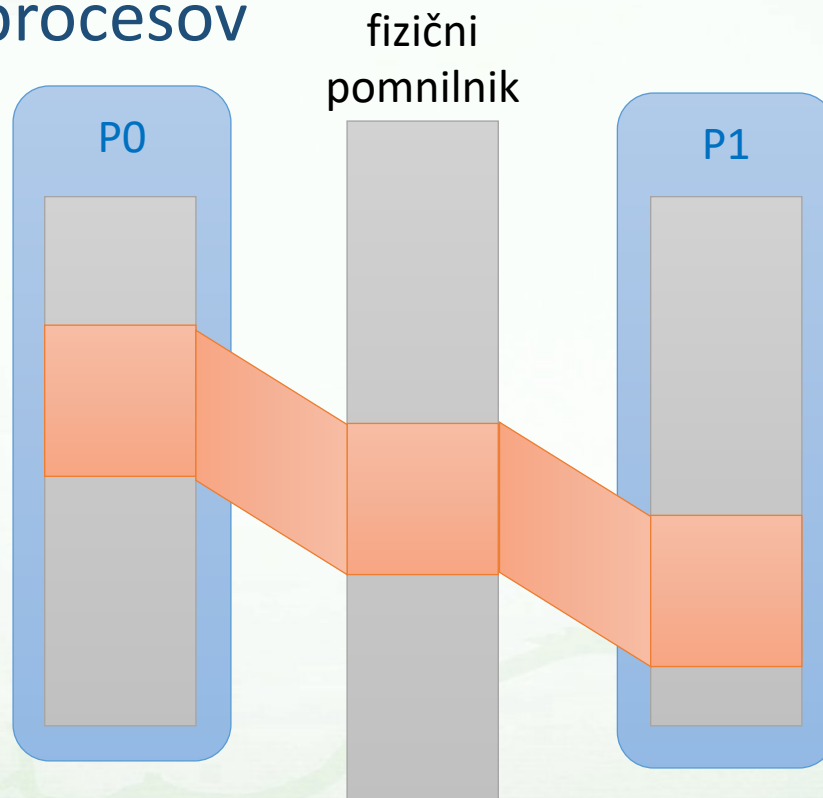
# Prenašanje sporočil

- Praktične izvedbe prenašanja sporočil
  - signali (Unix)
  - anonimne in imenovane cevi (pipes)
  - sporočilne vrste (message queues)
    - možne prioritete sporočil
    - SysV, POSIX
  - vtičnice
  - itd.



# Deljeni pomnilnik

- Deljeni pomnilnik
  - pomnilnik, ki je skupen dvema ali več procesom
  - del fizičnega pomnilnika, preslikan v naslovni prostor procesov



# Deljeni pomnilnik

- Uporaba

- vzpostavitev

- storitev OS (sistemski klici)
    - dodajanje deljenega pomnilnika v naslovni prostor
    - konfiguracija preslikave naslovnega prostora

- komunikacija

- neposredno branje in pisanje v deljeni pomnilnik

- sprostitvev oz. ločitev

- po koncu uporabe deljeni pomnilnik sprostim

# Deljeni pomnilnik

- Uporaba

- prednosti

- hitrost
    - podatki so prejemniku na voljo takoj

- slabosti

- potrebna eksplicitna sinhronizacija
    - programerjeva odgovornost za komunikacijski protokol

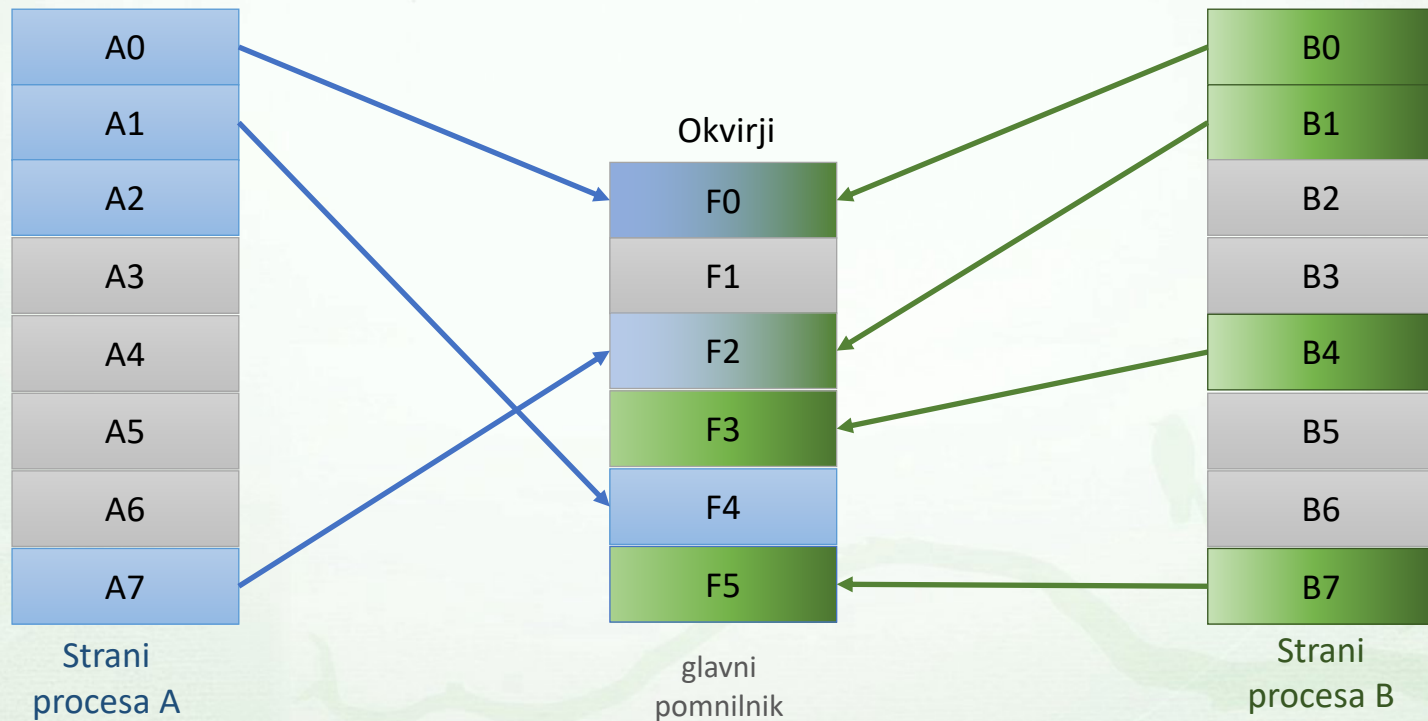


# Deljeni pomnilnik

- Kako sinhronizirati dostop?
  - preko mehanizmov v knjižnici pthreads
  - preko mehanizmov, podprtih s strani OS
  - preko sinhronega prenašanja sporočil
    - P1 podatke zapiše v deljeni pomnilnik
    - P1 pošlje sporočilo “ready” procesu P2
    - P2 prejme sporočilo in pošlje “ok” nazaj

# Deljeni pomnilnik

- Izvedba pri **ostranjevanju**
  - del navideznega prostora več procesov je preslikan v iste okvirje strani



# Deljeni pomnilnik

- Praktične izvedbe deljenega pomnilnika
  - System V API
  - POSIX API
  - pomnilniško preslikane datoteke
  - Android ashmem
  - niti: imajo skupen naslovni prostor



# Primerjava

- Način prenašanja podatkov
  - **prenašanje sporočil**
    - **kopiranje** podatkov med različnimi naslovnimi prostori
    - hitra vzpostavitev, počasna komunikacija
    - čas kopiranja podatkov
  - **deljeni pomnilnik**
    - **preslikava** vsebine pomnilnika med naslovnimi prostori
    - manj hitra vzpostavitev, hitra komunikacija
    - čas vzpostavitve preslikave
    - enkratna vzpostavitev, večkratna uporaba

# Primerjava

- Učinkovitost

Method	100 Byte Messages	1 Kilo Byte Messages
Unix Signals	--broken--	--broken--
zeromq	24,901 msg/s	22,679 msg/s
Internet sockets (TCP)	70,221 msg/s	67,901 msg/s
Domain sockets	130,372 msg/s	127,582 msg/s
Pipes	162,441 msg/s	155,404 msg/s
Message Queues	232,253 msg/s	213,796 msg/s
FIFOs (named pipes)	265,823 msg/s	254,880 msg/s
Shared Memory	4,702,557 msg/s	1,659,291 msg/s
Memory-Mapped Files	5,338,860 msg/s	1,701,759 msg/s

Vir: <https://github.com/goldsborough/ipc-bench>