

Operacijski sistemi

vaje 8

procesi

- strojna koda
- izvorna koda
- prevajanje
- izvršljiva datoteka
- proces
- večopravilni sistem
- razvrščevalnik

procesi

- identiteta procesa
 - PID, lastnik, skupina procesa
- okolje procesa
 - argumenti in okoljske spremenljivke
- kontekst procesa
 - stanje, prioriteta, registri, sklad, odprte datoteke, itd.

ukazi

- pidof, pgrep
- ps, top, pstree
- jobs, fg, bg
- kill, pkill, killall
- sleep
- wait
- nice, renice

navidezni imenik /proc/PID

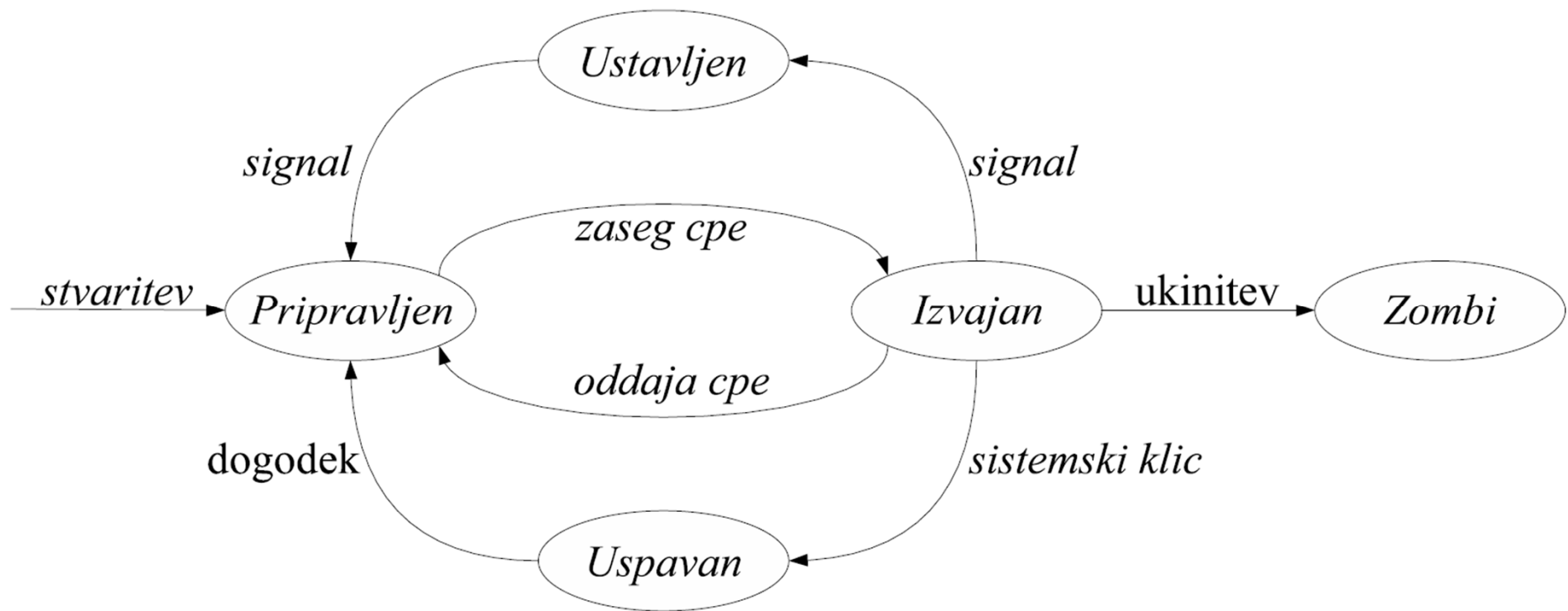
- cmdline
- cwd
- environ
- exe
- stat
- status
- fd
- task

naloge (1)

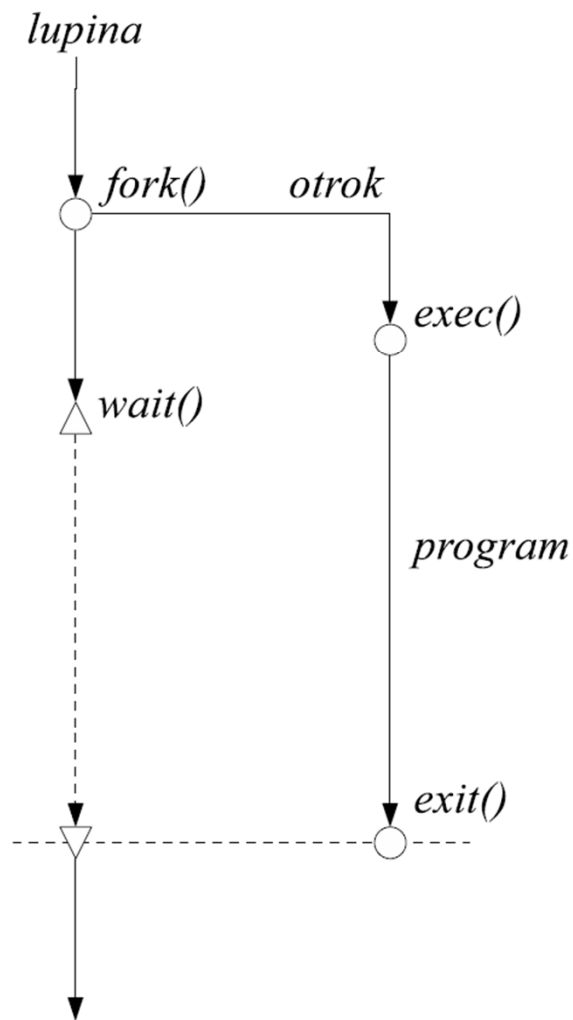
Prijavite se kot uporabnik `student` in kot uporabnik `administrator` v dve virtualni konzoli.

1. Izpišite PID-je terminalov `bash`.
2. Izpišite PID samo svojega terminala.
3. Izpišite vse procese na sistemu. Koliko jih je trenutno?
4. Izpišite:
 - a) samo svoje procese;
 - b) samo procese, ki so pognani iz trenutnega terminala.
5. Izpišite vse procese uporabnika `root` in njihove podatke izpišite v daljši obliki.
6. Iz ukaza `top` ubijte enega od svojih procesov (prej seznam omejite samo na svoje procese).
7. Izpišite drevo vseh podprocesov terminala, v katerem se nahajate.
8. Ustvarite cevovod `testfifo`. V eni konzoli poženite `cat >testfifo`, v drugi konzoli poženite `cat <testfifo | tee c.txt d.txt` in pogledajte datotečne deskriptorje vseh procesov v imeniku `/proc/`.
9. V imeniku `/proc/` pogledajte tudi okoljske spremenljivke obeh ukazov, status (v statusu poiščite PID očeta), lokacijo programa, ki je bil pognan, in argumente, ki so bili podani ob zagonu programa.

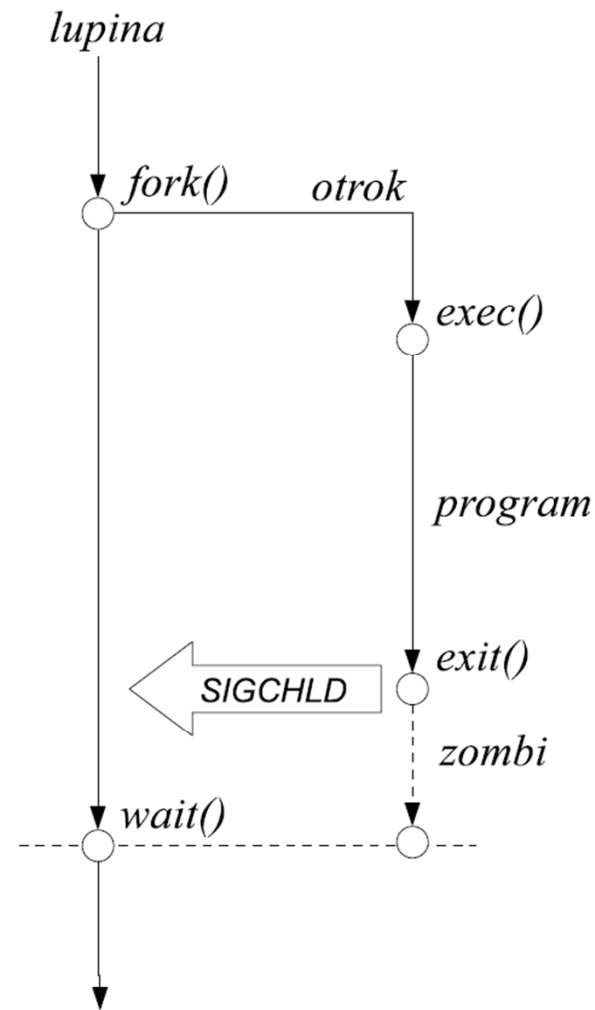
Diagram prehajanja stanj procesa



Zagon v ospredju in ozadju



a)



b)

signali

1	:	SIGHUP	:	exit	Hangup. Signalizira ukinitvev pripadajočega terminala.
2	:	SIGINT	:	exit	Prekinitev procesa (Ctrl+C).
3	:	SIGQUIT	:	core	Ukinitvev procesa (Ctrl+\).
4	:	SIGILL	:	core	Napačen ukaz procesorja.
8	:	SIGFPE	:	core	Izjema pri delu s števili v plavajoči vejici.
9	:	SIGKILL	:	exit	Brezpogojna ukinitvev procesa.
10	:	SIGUSR1	:	exit	Prvi uporabniški signal.
12	:	SIGUSR2	:	exit	Drugi uporabniški signal.
13	:	SIGPIPE	:	exit	Neuspeh pri pisanju v cev ali vtičnico.
14	:	SIGALRM	:	exit	Alarm.
15	:	SIGTERM	:	exit	Ukinitvev procesa.
17	:	SIGCHLD	:	ignore	Obvestilo staršu o ukinitvi otroka.
18	:	SIGCONT	:	ignore	Nadaljuje ustavljen proces.
19	:	SIGSTOP	:	stop	Brezpogojna (začasna) ustavitev procesa.
20	:	SIGTSTP	:	stop	(Začasna) ustavitev procesa (Ctrl+Z).
21	:	SIGTTIN	:	stop	Proces v ozadju skuša brati s terminala.
22	:	SIGTTOU	:	stop	Proces v ozadju skuša pisati na terminal.

naloge (2)

1. Iz konzole poženite program `gedit`. Kaj se zgodi s konzolo?
2. Začasno ustavite izvajanje procesa `gedit` in izpišite seznam trenutnih opravil.
3. Proces `gedit` naj nadaljuje z izvajanjem, tokrat v ozadju.
4. Kaj se zgodi s procesom `gedit`, če zapremo konzolo, iz katere smo ga pognali?
5. Kako bi preprečili zapiranje programa, če zapremo njegovega očeta?
6. Poženite naslednje procese v ozadju: `gedit`, `xeyes`, `xclock`. Proces `gedit`, ki teče v ozadju, naj se ponovno izvaja v ospredju.
7. Prekinite izvajanje procesa `gedit` iz druge konzole s pomočjo ukaza `kill`. Kakšen je izhodni status procesa `gedit`?
8. Iz druge konzole začasno ustavite izvajanje procesa `xeyes`. Pomagajte si signalom `STOP`.
9. Iz druge konzole ponovno poženite ustavljen proces `xeyes`.
10. Poženite ukaz `tree / >izpis.txt` z najvišjo prioriteto. Koliko hitreje se bo ukaz izvedel, če je to edini proces, ki teče na procesorju?
11. Napišite program, ki izvaja neskončno zanko. Iz konzole ga dvakrat poženite v ozadju. Spreminjajte prioriteto procesov in opazujte delež procesorskega časa.