

Kontrola poslova



Ukoliko ste u naslovu prepoznali prijevod izraza "*job control*", možda i ne trebate pročitati ovaj članak jer ste upoznati s tematikom. Ipak, tema je samo na prvi pogled trivijalna, pa je dobro da "bacite pogled".

Pod kontrolom zadataka, odnosno poslova, podrazumijevamo pokretanje procesa u korisničkoj ljusci (*shellu*), njihovo zaustavljanje i stavljanje izvršavanja u pozadinu.

Pojednostavljeno rečeno, svaki proces kojeg pokrenemo iz shella dobija svoj broj (PID, process id) i preuzima kontrolu nad standardnim ulazom (STDIN). Ukoliko je riječ o procesu koji dugo traje, možemo ga staviti u pozadinu i nastaviti s radom. Kako to možemo učiniti? Prvo je potrebno zaustaviti proces s tipkama **Ctrl+Z**. Za razliku od kombinacije **Ctrl+C**, koja šalje signal **SIGINT**, **Ctrl+Z** šalje **SIGTSTP**. Ovaj signal zaustavlja izvršavanje procesa, ali ga ne "ubija" (u smislu zaustavljanja i brisanja iz procesne tablice i memorije). Primjer:

```
server$ find / -name datoteka.txt -print
<Ctrl+Z>
[1]+  Stopped                  find /
server$ ps -ef | grep find
root    17037  7095  0 23:12 pts/8    00:00:00 find /
server$ ps -e -o pid,comm,state| grep 17037
17037 find
```

Kad zaustavimo proces, njegov se status mijenja u "T", što znači "Stopped". Najčešće vidamo status "R" (Running). Za ostale statuse pogledajte *manual* naredbe "ps".

Sada imamo zaustavljen proces, preostaje da ga stavimo u pozadinu, gdje će se on nastaviti izvršavati. To radimo pomoću naredbe "bg":

```
server$ bg
[1]+ find / -name datoteka.txt -print &
```

Proces koji je otvorila naredba find nastavlja se izvršavati u pozadini, a ukoliko, kao u primjeru, niste preusmjerili izlaz u datoteku, moguće je da vam se u terminalu pojavljuje ispis te naredbe (ukoliko ga ima). Kada naredba završi sa izvršavanjem, dobit ćete poruku:

```
[1]+ Done                  find / -name datoteka.txt -print
```

Kako naredba find (ili bilo koja druga koju ste pokrenuli) može potrajati s izvršavanjem, možete je "vratiti natrag" u prednji plan (*foreground*) naredbom **fg**:

```
server$ fg
find / -name datoteka.txt -print
...
```

Sada možete prekinuti izvršavanje naredbe sa Ctrl+C ili sačekati da obavi posao, što vam već u toj situaciji treba.

Možete imati istovremeno nekoliko pozadinskih procesa. Sve procese na jednak način možete poslati u pozadinu, a njihov popis se može vidjeti s naredbom "jobs":

```
server$ jobs
[1]  Stopped          find /
[2]  Stopped          vim /etc/hosts.deny
[3]  Running          sleep 2000 &
[4]- Running         sleep 3000 &
[5]+ Running         sleep 4000 &
```

Naredbe **bg** i **fg** (zapravo, radi se o *shell builtin* funkcijama) primaju argumente, koji su brojevi procesa u pozadini. Ukoliko želite vratiti u prvi plan proces broj 3, napravite:

```
server$ fg %3
sleep 2000
Ctrl+C
```

Ovakvo postupanje s procesima je zapravo rijetko. Obično se procesi pokreću odmah u pozadini. Ovo je tipično kada pokrećete neki GUI program iz terminalskog emulatora:

```
server$ xclock &
[1] 27491
```

Da xclock nije pokrenut u pozadini, blokirao bi terminal, a ovako se pokreće, ali i vraća vam prompt i punu kontrolu nad terminalom.

Važno je napomenuti da svo procesi pokrenuti iz shella se gase zajedno s tim *shellom*. Ovo znači da svi procesi u pozadini ne mogu preživjeti gašenje *shella*, odnosno *logout*. Da biste ovo spriječili, koristi se naredba "nohup":

```
server$ nohup naredba_koja_dugo_traje &
```

Ovako pokrenuta naredba preživjet će *logout*, ali ćete izgubiti kontrolu nad njenim ulazom i izlazom. Zato preusmjerite izlaz u datoteku, kako biste znali u kojem stadiju izvršavanja se naredba nalazi.

Neke funkcije možete skoro u potpunosti zamijeniti naredbom "kill". Iako je većini sistemaca u glavi da je *kill* "naredba za ubijanje procesa", istina je malo složenija. Naredba *kill* šalje signale procesima, a oni u ovisnosti o tome o kojem se signalu radi prestaju s radom, zaustavljaju se (pauziraju) ili rade neku drugu akciju.

Naredba *bg* jednostavno šalje procesu signal *SIGTSTP*:

```
server$ jobs
[1]+  Running          sleep 1000 &
po:/home/csk/zolja# kill -TSTP 27491

[1]+  Stopped          sleep 1000
```

Ako procesu pošaljemo signal *SIGCONT*, on će nastaviti s radom:

```
server$ kill -CONT 27491
server$ jobs
[1]+  Running                  sleep 1000 &
```

Kako je i naredba *kill* zapravo ugrađena u ljusku, možemo koristiti i ovakav oblik:

```
server$ type kill
kill is a shell builtin
server$ kill -TSTP %1
[1]+  Stopped                  sleep 1000
server$ kill -CONT %1
server$ jobs
[1]+  Running                  sleep 1000 &
```

Popis svih signala (uključujući i numerički oblik) možete naći ukoliko upotrijebite naredbu *kill -l*:

```
server$ kill -l
 1) SIGHUP    2) SIGINT    3) SIGQUIT    4) SIGILL    5) SIGTRAP
 6) SIGABRT   7) SIGBUS    8) SIGFPE    9) SIGKILL  10) SIGUSR1
11) SIGSEGV  12) SIGUSR2  13) SIGPIPE  14) SIGALRM  15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD  18) SIGCONT  19) SIGSTOP  20) SIGTSTP
21) SIGTTIN  22) SIGTTOU  23) SIGURG   24) SIGXCPU  25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF  28) SIGWINCH 29) SIGIO    30) SIGPWR
31) SIGSYS    34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```

Zdravko Rašić

uto, 2013-04-30 23:38 - Zdravko Rašić **Kuharice:** [Linux](#) [1]

Kategorije: [Operacijski sustavi](#) [2]

Vote: 0

No votes yet

Source URL: <https://sysportal.carnet.hr/node/1260>

Links

[1] <https://sysportal.carnet.hr/taxonomy/term/17>

[2] <https://sysportal.carnet.hr/taxonomy/term/26>

