

Systemd - međuovisnosti unita

U prošlom smo nastavku objasnili što su units (objekti kojima systemd upravlja) i targets (ciljevi koje treba postići). Vrijeme je da se pozabavimo redoslijedom podizanja i spuštanja servisa. Zapravo, trebali bi se navikavati na novu terminologiju: radi se o međuovisnostima unita, jedinica. Sistemci stare škole po navici brinu samo o servisima, smatrajući da se ostali detalji podrazumijevaju. Ali jedinica ne mora biti samo servis, jer systemd ima načina da brine o detaljima kojima se stari init nije zamarao.

Evo odmah i primjera. Servis acpid ovisi o svom socketu, pa će očekivati da je socket spreman. Zato ćemo u konfiguracijskoj datoteci `/lib/systemd/system/acpid.service` pronaći ovakav zapis:

```
Requires=acpid.socket
```

Dakle ne ovise samo servisi jedni o drugima, već i o nekim temeljnim sistemskim funkcijama. Systemd je sve "razmrvio" na sastavnice: na vrhu je cilj (`.target`) koji želimo postići, a do njega se dolazi pomoću različitih objekata: servisa (`.service`) i jedinica nižeg reda, poput socketa iz našeg primjera (`.socket`), mountpointa (`.mount`), putanje (`.path`), uređaja (`.device`) itd.

Pokušat ćemo definiranje međuovisnosti u konfiguraciji objasniti što jednostavnije. Tri su osnovne međuovisnosti:

```
Requires=ime-unita
Wants=ime-unita
After=ime-unita
```

Ako je ovisnost obavezna, stavljamo

```
Requires=neki-unit.
```

Ako je samo poželjna, tada

```
Wants=neki-unit
```

Ako k tome još želimo naglasiti da je važan vremenski redoslijed, dodat ćemo

```
After=neki-unit.
```

U čemu je poanta? Zar nije dovoljno reći da se zahtijeva da je neki unit već spreman? Jednostavno, ako ne stavimo `After` tada će se servisi pokrenuti paralelno, a onda će uspostavljati međusobnu komunikaciju. U slučaju kada neki servis neće raditi ako neki drugi uvjet, odnosno unit nije već pripremljen, tada moramo staviti `After`. Na primjer, ako koristimo web aplikaciju koja se spaja na bazu podataka, logično je da se najprije pokrene baza, pa web server i aplikacija. Web servis će raditi i bez baze, ali aplikacija će javljati grešku jer ne može do podataka. Da se to spriječi, iskoristit ćemo `After` kako bi bili sigurni da će baza biti spremna prije nego se pokrene web servis.

Vrijeme je da upoznamo konfiguracijske datoteke kojima se opisuju jedinice. Obratite pažnju na izraz "opisuje". Tu se systemd razlikuje od SysVinita koji je koristio izvršne skripte za pokretanje i gašenje servisa. Evo datoteke koja opisuje servis `anacron`.

```
[Unit]
Description=Run anacron jobs
After=time-sync.target
ConditionACPower=true

[Service]
ExecStart=/usr/sbin/anacron -dsq
IgnoreSIGPIPE=false

[Install]
WantedBy=multi-user.target
```

Na prvi je pogled jasno da se opis sastoji od tri dijela. Sekcija Unit je obavezna i naći ćemo je u svim datotekama koje opisuju jedinice. Sekcija Service je, logično, prisutna samo ako se opisuje servis. Sekcija Install također nije obavezna i koristi je samo `systemctl enable/disable` naredba. U primjeru anacron servisa, nakon opisa slijedi zahtjev da se prije pokretanja anacrona obavi sinkronizacija vremena, kako bi dobili točno vrijeme. I ostatak ne izgleda previše zagonetan. Pod Service je putanja izvršne datoteke s parametrima za pokretanje, a u Install sekciji se kaže da ovu jedinicu traži `multi-user.target`, nekadašnji `runlevel 3`.

No ne treba se prerano radovati jednostavnosti naših primjera. Konfiguraciju je moguće jako zakomplicirati, još je mnogo stavki koje će s vremenom trebati usvojiti. Ali to ćemo ostaviti za neku drugu priliku. Zasad je dovoljno upoznati osnovne pojmove.

čet, 2017-08-31 18:47 - Aco Dmitrović **Kuharice:** [Linux](#) [1]

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

Vote: 0

No votes yet

story_tag: [Linux](#) [3]

[systemd](#) [4]

[units](#) [5]

Source URL: <https://sysportal.carnet.hr/node/1754>

Links

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

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

[3] <https://sysportal.carnet.hr/taxonomy/term/119>

[4] <https://sysportal.carnet.hr/taxonomy/term/120>

[5] <https://sysportal.carnet.hr/taxonomy/term/121>