

Współczesne systemy komputerowe

Start systemu, zarządzanie usługami

Start systemu

- System operacyjny Linux startuje w następujących krokach
 - **BIOS** (*Basic Input/Output System*) lub **UEFI** (*Unified Extensible Firmware Interface*)
 - **Boot Manager** (*GRUB2*)
 - **Jądro systemu** (*kernel*)
 - **Startowy ramdysk** (*Initial RAM File System, Initial RAM Disk*)
 - **Menadżer startu** systemu i usług (*systemd*, dawniej *init*)
 - Kolejne procesy i usługi systemu (uruchamia je demon *systemd*)

GRUB

- Jest to program, którego część rezyduje w MBR dysku lub sektorach startowych partycji, reszta na początkowych sektorach dysku i w katalogu **/boot**
- GRUB potrafi załadować jądro systemu, plik **/boot/vmlinuz**, który faktycznie jest linkiem do **/boot/vmlinuz-version**
- GRUB potrafi uruchamiać też inne systemy operacyjne, m. in. z rodziny Windows
- Wyświetl plik konfiguracyjny GRUB'a

```
root@debian:~# cat /boot/grub/grub.cfg | grep timeout
set timeout=30
if [ x$feature_timeout_style = xy ] ; then
  set timeout_style=menu
  set timeout=5
# Fallback normal timeout code in case the timeout_style feature is
set timeout=5
```

- Tego poliku konfiguracyjne nie powinno się edytować ręcznie, zmiany wprowadza się w **/etc/default/grub**
- Sprawdź i zmień czas oczekiwania (z **5** na **10** sekund, użyj edytora **nano**) na wybór systemu operacyjnego w GRUB'ie

```
root@debian:~# cat /etc/default/grub | grep TIMEOUT
GRUB_TIMEOUT=5
```

```
root@debian:~# nano /etc/default/grub
```

- Po wprowadzeniu zmian wygeneruj nowe pliki konfiguracyjne

```
root@debian:~# update-grub
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-4.9.0-6-686
Found initrd image: /boot/initrd.img-4.9.0-6-686
Found linux image: /boot/vmlinuz-4.9.0-4-686
Found initrd image: /boot/initrd.img-4.9.0-4-686
```

```
Found linux image: /boot/vmlinuz-4.9.0-3-686
Found initrd image: /boot/initrd.img-4.9.0-3-686
Done
```

```
root@debian:~# cat /boot/grub/grub.cfg | grep timeout
set timeout=30
if [ x$feature_timeout_style = xy ] ; then
  set timeout_style=menu
  set timeout=10
# Fallback normal timeout code in case the timeout_style feature is
set timeout=10
```

- Zrestartuj system, zwróć uwagę na czas oczekiwania na wybór systemu

```
root@debian:~# systemctl reboot
```

- Zaznacz *Debian GNU/Linux* i przejdź do trybu edycji, wciśnij [e]
- Na końcu linii `linux /vmlinuz-...` dopisz `init=/bin/bash` a następnie klawisze [CTRL X] (start systemu)
- W taki sposób można uruchomić powłokę **bash** z prawami użytkownika *root*, można zmienić w ten sposób zapomniane hasło administratora, należy jednak wcześniej zamontować system plików z możliwością zapisu

```
root@(none):/# whoami
root
```

```
root@(none):/# mount -o remount,rw,sync /
```

```
root@(none):/# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

- Zamontuj system plików tylko do odczytu i wyłącz komputer (jeśli wirtualny, zamknij okno systemu gościa, nie zapisuj stanu maszyny)

```
root@(none):/# mount -o remount,ro /
```

- Dlatego w systemach produkcyjnych zależy zabezpieczać **BIOS/UEFI** i loader systemu hasłem

systemd

- Menadżerem startu systemu i usług jest demon **systemd**
- W stosunku do skryptów **SysVinit** lub **BSDinit** ma większe możliwości:
 - możliwość równoległego uruchamiania usług
 - uruchamianie serwisów za pomocą aktywacji gniazda lub na żądanie
 - śledzenie działania procesów przy pomocy grup kontrolnych *cgroups*
 - wsparcie dla zachowywania i przywracania stanu usług w systemie
 - utrzymanie punktów montowania i automatycznego montowania w systemie
- **systemd** posługuje się tzw. *unitami*, wyświetl typy unitów

```
root@debian:~# systemctl -t help
Available unit types:
service
socket
busname
target
device
mount
automount
swap
timer
path
slice
scope
```

- Podstawowe typy unitów

*.service	serwisy i usługi systemowe
*.timer	aktywacja zadań w oparciu o czas
*.socket	aktywacja usług w oparciu o port (systemd nasłuchuje port)
*.target	podobne do klasycznych poziomów pracy, grupowanie unitów w logiczną funkcjonalność
*.path	monitorowanie ścieżki w celu aktywacji zadania
*.mount	montowanie systemów plików (np. /boot , /tmp)
*.automount	automontowanie systemów plików (np. /proc , /sys)
*.device	pliku urządzeń

- Wyświetl wszystkie dostępne unity związane z usługami (zakończenia działania polecenia wymaga wciśnięcia klawisza [q])

```
root@debian:~# systemctl list-unit-files --type=service
```

```
UNIT FILE                                STATE
accounts-daemon.service                 enabled
alsa-restore.service                   static
alsa-state.service                      static
alsa-utils.service                     masked
anacron.service                        enabled
apt-daily-upgrade.service              static
apt-daily.service                      static
autovt@.service                        enabled
avahi-daemon.service                   enabled
bluetooth.service                     enabled
bootlogd.service                      masked
bootlogs.service                      masked
bootmisc.service                      masked
checkfs.service                       masked
checkroot-bootclean.service            masked
checkroot.service                     masked
colord.service                         static
configure-printer@.service             static
console-getty.service                 disabled
console-setup.service                 enabled
container-getty@.service              static
cron.service                          enabled
...
```

- Każdy unit może być w jednym ze stanów

enabled	uruchamiany po starcie systemu
disabled	nieuruchamiany po starcie systemu, ale może być załadowany jako zależność
static	może być uruchomiony przez inny unit, nie może być uruchomiony ręcznie
masked	mocniejsze niż <i>disabled</i> , nie może być uruchomiony jako zależność, może być uruchomiony po <i>odmaskowaniu</i>
generate d	wygenerowany automatycznie na podstawie skryptów <i>/etc/init.d</i>

Poziomy pracy

- Stan systemu operacyjnego definiują tzw. **poziomy pracy**

poweroff.target	zatrzymywanie systemu: (poweroff, shutdown -h)
rescue.target	tryb awaryjny, minimalny zestaw usług
multi-user.target	wielu użytkowników, bez interfejsu graficznego
graphical.target	wielu użytkowników, z interfejsem graficznym
reboot.target	restart systemu (reboot, shutdown -r)

- Sprawdź domyślny poziom pracy systemu

```
root@debian:~# systemctl get-default
graphical.target
```

- Zmień poziom pracy systemu

```
root@debian:~# systemctl isolate multi-user.target
```

- Zaloguj się w konsoli tekstowej jako *root*
- Sprawdź poziom pracy systemu, zmień poziom pracy na graficzny

```
root@debian:~# systemctl isolate graphical.target
```

- Kombinacją klawiszy [**CTRL ALT F2**] przełącz się na konsolę tekstową, zaloguj się i sprawdź działanie skrótów [**ALT F3**] do [**ALT F6**]
- Kombinacją [**CTRL ALT F1**] wróć do środowiska graficznego
- Zmień domyślny poziom pracy systemu, zrestartuj maszynę

```
root@debian:~# systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target → /lib/systemd/system/multi-user.target.
```

```
root@debian:~# systemctl get-default
multi-user.target
```

```
root@debian:~# systemctl reboot
```

- Ponownie zmień domyślny poziom pracy systemu i zrestartuj maszynę

```
root@debian:~# systemctl set-default graphical.target
Created symlink /etc/systemd/system/default.target →
/lib/systemd/system/graphical.target.
```

```
root@debian:~# systemctl get-default
graphical.target
```

```
root@debian:~# systemctl reboot
```

Zarządzanie usługami

- Wyświetl załadowane usługi systemu

```
root@debian:~# systemctl list-units --type=service
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
alsa-restore.service	loaded	active	exited	Save/Restore Sound Card S
avahi-daemon.service	loaded	active	running	Avahi mDNS/DNS-SD Stack
colord.service	loaded	active	running	Manage, Install and Gener
console-setup.service	loaded	active	exited	Set console font and keym
cron.service	loaded	active	running	Regular background progra
cups-browsed.service	loaded	active	running	Make remote CUPS printers
cups.service	loaded	active	running	CUPS Scheduler
dbus.service	loaded	active	running	D-Bus System Message Bus
gdm.service	loaded	active	running	GNOME Display Manager
getty@tty2.service	loaded	active	running	Getty on tty2
getty@tty3.service	loaded	active	running	Getty on tty3
getty@tty4.service	loaded	active	running	Getty on tty4
getty@tty5.service	loaded	active	running	Getty on tty5
getty@tty6.service	loaded	active	running	Getty on tty6
keyboard-setup.service	loaded	active	exited	Set the console keyboard
kmod-static-nodes.service	loaded	active	exited	Create list of required s
minissdpd.service	loaded	active	running	keep memory of all UPnP d
ModemManager.service	loaded	active	running	Modem Manager
networking.service	loaded	active	exited	Raise network interfaces
NetworkManager-wait-online.service	loaded	active	exited	Network Manager Wait On
NetworkManager.service	loaded	active	running	Network Manager
...				

- Status usługi, kolumna **ACTIVE**:
 - **active** - usługa działa
 - **inactive** - usługa nie działa, często z powodu błędu
- Kolumna **SUB**:
 - **running** - usługa jest uruchomiona
 - **exited** - usługa zakończona sukcesem
 - **waiting** - usługa czeka na zdarzenie
- Sprawdź status usługi **cron**

```
root@debian:~# systemctl status cron
```

```
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: ena
   Active: active (running) since Thu 2018-03-15 07:06:55 EDT; 1h 52min ago
```

```

Docs: man:cron(8)
Main PID: 334 (cron)
Tasks: 1 (limit: 4915)
CGroup: /system.slice/cron.service
└─334 /usr/sbin/cron -f

Mar 15 07:06:55 debian systemd[1]: Started Regular background program processing
Mar 15 07:06:55 debian cron[334]: (CRON) INFO (pidfile fd = 3)
Mar 15 07:06:55 debian cron[334]: (CRON) INFO (Running @reboot jobs)
Mar 15 07:17:01 debian CRON[1039]: pam_unix(cron:session): session opened for us
Mar 15 07:17:01 debian CRON[1040]: (root) CMD ( cd / && run-parts --report /et
Mar 15 07:30:01 debian CRON[1155]: pam_unix(cron:session): session opened for us
Mar 15 07:30:01 debian CRON[1156]: (root) CMD ([ -x /etc/init.d/anacron ] && if
Mar 15 08:17:01 debian CRON[2095]: pam_unix(cron:session): session opened for us
Mar 15 08:17:01 debian CRON[2096]: (root) CMD ( cd / && run-parts --report /et

```

- Wyłącz usługę i sprawdź jej status

```

root@debian:~# systemctl disable cron
Synchronizing state of cron.service with SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable cron

```

```

root@debian:~# systemctl list-unit-files --type=service

```

```

UNIT FILE                                STATE
...
cron.service                             disabled
...

```

```

root@debian:~# systemctl status cron

```

```

• cron.service - Regular background program processing daemon
  Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: ena
  Active: inactive (dead) since Thu 2018-03-15 09:00:17 EDT; 18s ago
  Docs: man:cron(8)
  Process: 334 ExecStart=/usr/sbin/cron -f $EXTRA_OPTS (code=killed, signal=TERM
  Main PID: 334 (code=killed, signal=TERM)

Mar 15 07:06:55 debian cron[334]: (CRON) INFO (pidfile fd = 3)
Mar 15 07:06:55 debian cron[334]: (CRON) INFO (Running @reboot jobs)
Mar 15 07:17:01 debian CRON[1039]: pam_unix(cron:session): session opened for us
Mar 15 07:17:01 debian CRON[1040]: (root) CMD ( cd / && run-parts --report /et
Mar 15 07:30:01 debian CRON[1155]: pam_unix(cron:session): session opened for us
Mar 15 07:30:01 debian CRON[1156]: (root) CMD ([ -x /etc/init.d/anacron ] && if
Mar 15 08:17:01 debian CRON[2095]: pam_unix(cron:session): session opened for us
Mar 15 08:17:01 debian CRON[2096]: (root) CMD ( cd / && run-parts --report /et
Mar 15 09:00:17 debian systemd[1]: Stopping Regular background program processin
Mar 15 09:00:17 debian systemd[1]: Stopped Regular background program processing

```

- Przywróć autostart usługi

```

root@debian:~# systemctl enable cron
Synchronizing state of cron.service with SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable cron

```