Chapter 02. 응용 프로그램 다루기

데몬 서비스 관리하기

Copyright **FASTCAMPUS** Corp. All Rights Reserved

데몬 서비스

- 서비스 관리하기 (service, systemctl)
- 서비스 로그 살펴보기 (journalctl)
- 나만의 서비스 만들기

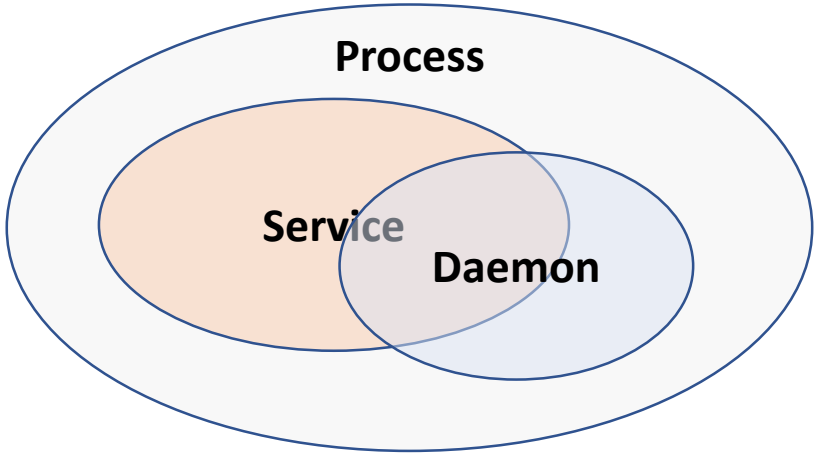
Copyright **FASTCAMPUS** Corp. All Rights Reserved

데몬이란(daemon)

사용자가 직접적으로 제어하지 않고, 백그라운드에서 돌면서 여러 작업을 하는 프로그램

서비스란(service)

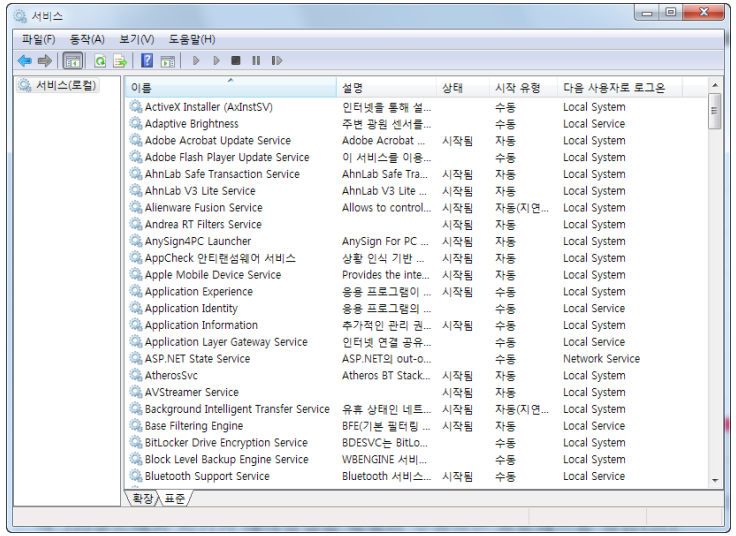
주로 서버/클라이언트 모델에서 출발하여, 사용자의 요청에 응답하는 프로그램 (주로 데몬 형태로 구동됨)



사용예)

- 웹서버 - httpd
- 파일서버 - ftpd
- 웹프록시 - squid
- 시큐어쉘(원격터미널) - sshd
- 시스템로깅 - syslogd, rsyslogd
- 프린터데몬 - cupsd, lpd
- 네트워크서비스 - inetd, xinetd

그 외 다수...



Copyright FASTCAMPUS Corp. All Rights Reserved

우분투의 데몬

SystemV 명령어 (service 를 통한 확인)

사실상 우분투16/18 에서는 (기본적으로는)
SysV 의 service를 사용하지 않음.

내부적으로는 systemctl 로 호출됨.

- service --status-all

- service <daemon-name> status
- service <daemon-name> start
- service <daemon-name> stop
- service <daemon-name> restart

```
user1@user1-VirtualBox:~$ service --status-all
 [ + ]  acpid
 [ - ]  alsa-utils
 [ - ]  anacron
 [ + ]  apparmor
 [ + ]  apport
 [ + ]  avahi-daemon
 [ - ]  bluetooth
 [ - ]  bootmisc.sh
 [ - ]  brltty
 [ - ]  checkfs.sh
 [ - ]  checkroot-bootclean.sh
 [ - ]  checkroot.sh
 [ - ]  console-setup.sh
 [ + ]  cron
 [ + ]  cups
 [ + ]  cups-browsed
 [ + ]  dbus
 [ - ]  dns-clean
 [ + ]  gdm3
 [ + ]  grub-common
 [ - ]  hostname.sh
 [ - ]  hwclock.sh
 [ - ]  irqbalance
 [ + ]  kerneloops
 [ - ]  keyboard-setup.dpkg-bak
 [ - ]  keyboard-setup.sh
 [ - ]  killprocs
 [ + ]  kmod
 [ - ]  lightdm
 [ - ]  mountall-bootclean.sh
 [ - ]  mountall.sh
 [ - ]  mountdevsubfs.sh
 [ - ]  mountkernfs.sh
 [ - ]  mountnfs-bootclean.sh
 [ - ]  mountnfs.sh
 [ + ]  network-manager
 [ + ]  networking
 [ - ]  ondemand
 [ - ]  plymouth
 [ - ]  plymouth-log
 [ - ]  pppd-dns
 [ + ]  procps
 [ - ]  rc.local
 [ + ]  resolvconf
 [ - ]  rsync
 [ + ]  rsyslog
 [ - ]  saned
 [ - ]  sendsigs
 [ + ]  speech-dispatcher
 [ - ]  spice-vdagent
 [ - ]  thermald
 [ + ]  udev
 [ + ]  ufw
 [ - ]  umountfs
 [ - ]  umountnfs.sh
 [ - ]  umountroot
 [ + ]  unattended-upgrades
 [ + ]  urandom
 [ - ]  uuidd
 [ + ]  whoopsie
 [ - ]  x11-common
```

Copyright FASTCAMPUS Corp. All Rights Reserved

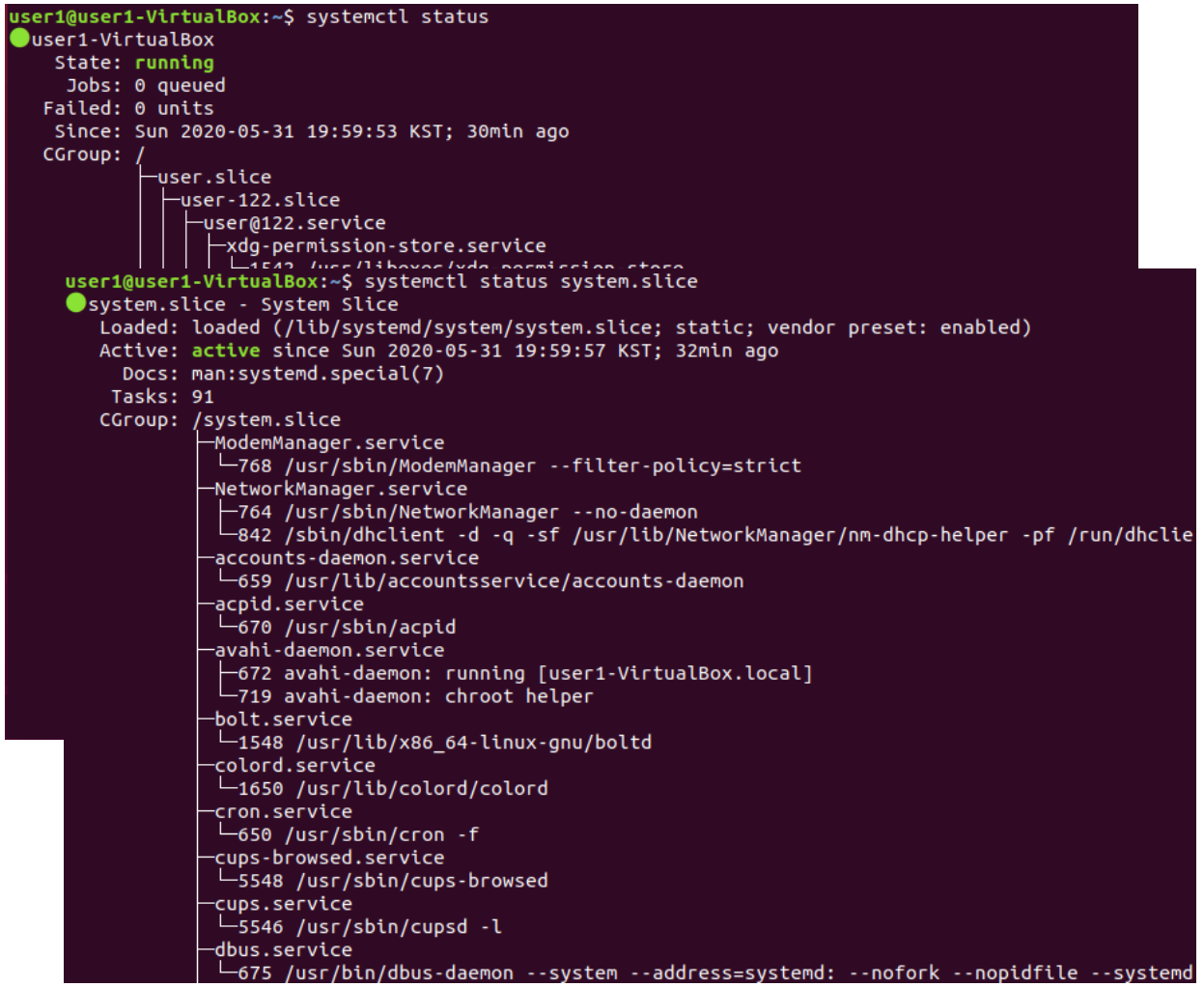
우분투의 데몬

systemd 를 사용하는 우분투의 systemctl 을 통한 서비스 확인

- systemctl status

- systemctl status <daemon-name>
- systemctl start <daemon-name>
- systemctl stop <daemon-name>
- systemctl restart <daemon-name>

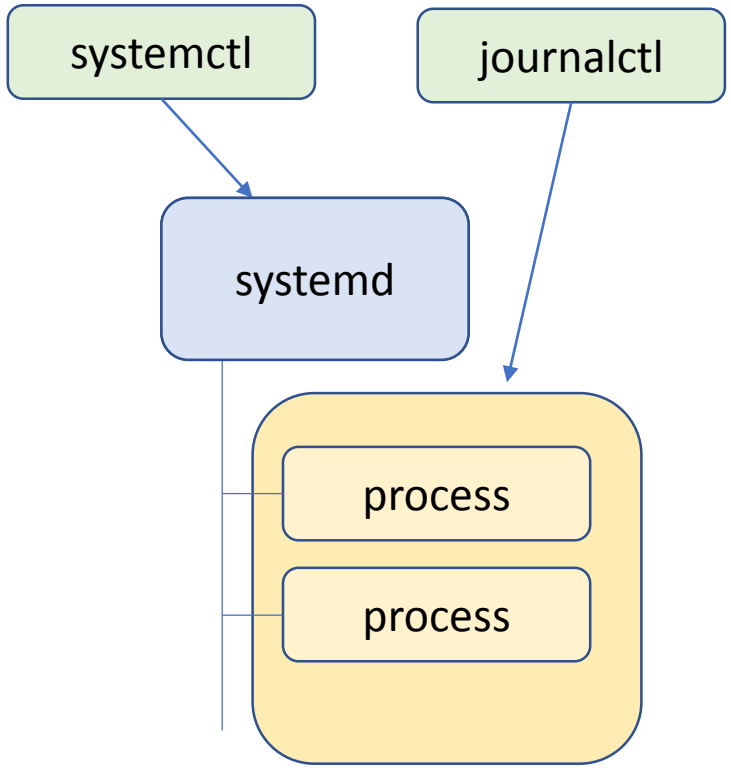
정확히는 <daemon-name>
⇒ daemon-name.service



Copyright FASTCAMPUS Corp. All Rights Reserved

Systemd 를 통한 서비스 관리

- systemd 는 대체 어디에?
  - /sbin/init -> /lib/system/systemd
- systemd 의 사용 이유
  - 프로세스의 자동 시작
  - 프로세스간의 의존성 관리
  - 프로세스의 갑작스런 종료에 대응
  - 부팅 옵션(런레벨) 에 따른 다른 프로세스 구동



Copyright FASTCAMPUS Corp. All Rights Reserved

Systemd 디렉토리 구조

- 시스템 서비스 디렉토리
  - /lib/systemd/system
  - 런레벨에 따른 타겟 서비스 목록
    - /lib/systemd/system/default.target
    - /lib/systemd/system/runlevel?.target
- 심볼릭 링크(/etc -> /lib) 를 통한 구현체
  - /etc/systemd/system/*.service

```
user1@user1-VirtualBox:~$ ls -l /lib/systemd/system/runlevel*.target
lrwxrwxrwx 1 root root 15  5월  3 20:30 /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13  5월  3 20:30 /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17  5월  3 20:30 /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17  5월  3 20:30 /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17  5월  3 20:30 /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16  5월  3 20:30 /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13  5월  3 20:30 /lib/systemd/system/runlevel6.target -> reboot.target
```

```
user1@user1-VirtualBox:~$ ls -ald /etc/systemd/system/*.target*
drwxr-xr-x 2 root root 4096  2월 27  2019 /etc/systemd/system/bluetooth.target.wants
drwxr-xr-x 2 root root 4096  2월 27  2019 /etc/systemd/system/default.target.wants
drwxr-xr-x 2 root root 4096  4월 17 23:54 /etc/systemd/system/final.target.wants
drwxr-xr-x 2 root root 4096  2월 27  2019 /etc/systemd/system/getty.target.wants
drwxr-xr-x 2 root root 4096  5월 24 04:11 /etc/systemd/system/graphical.target.wants
drwxr-xr-x 2 root root 4096  6월  2 00:56 /etc/systemd/system/multi-user.target.wants
drwxr-xr-x 2 root root 4096  2월 27  2019 /etc/systemd/system/network-online.target.wants
drwxr-xr-x 2 root root 4096  5월 24 04:16 /etc/systemd/system/paths.target.wants
drwxr-xr-x 2 root root 4096  2월 27  2019 /etc/systemd/system/printer.target.wants
drwxr-xr-x 2 root root 4096  4월 17 23:54 /etc/systemd/system/sockets.target.wants
drwxr-xr-x 2 root root 4096  5월 24 04:15 /etc/systemd/system/spice-vdagentd.target.wants
drwxr-xr-x 2 root root 4096  5월 24 04:18 /etc/systemd/system/sysinit.target.wants
drwxr-xr-x 2 root root 4096  5월 24 04:15 /etc/systemd/system/timers.target.wants
```

Copyright FASTCAMPUS Corp. All Rights Reserved

Systemd 유닛 종류

- 시스템디의 유닛 종류
    - service, socket, device, mount, automount, swap, target, path, timer, slice, scope
        - /lib/systemd/system/*.service
        - /lib/systemd/system/*.socket
        - /lib/systemd/system/*.mount
        - …

Copyright **FASTCAMPUS** Corp. All Rights Reserved

시스템 컨트롤 명령어

Systemctl 을 통한 다양한 데몬/서비스 확인

- 실행 중인 서비스 목록 확인
  - systemctl list-units
    - 서비스 확인
      - systemctl list-units --type=service
      - 예) systemctl list-units --type=service --state=running (상태=failed, active, running)
    - 부팅 시 선택 가능한 타겟 옵션 확인
      - systemctl list-units --type=target
- 부팅 시 기본 옵션 (현재 runlevel 확인)
  - systemctl get-default
- 부팅 시 기본 옵션 변경 (runlevel 변경)
  - systemctl set-default <target>
  - 예) sudo systemctl set-default multi-user.target
    부팅 시 GUI 로 로그인하지 않고 CLI로 사용

```
user1@user1-VirtualBox:~$ systemctl list-units --type=target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
cryptsetup.target      loaded active active Local Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target  loaded active active Network is Online
network-pre.target     loaded active active Network (Pre)
network.target         loaded active active Network
nss-lookup.target      loaded active active Host and Network Name Lookups
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
sound.target           loaded active active Sound Card
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
time-sync.target       loaded active active System Time Synchronized
timers.target          loaded active active Timers
```

Copyright FASTCAMPUS Corp. All Rights Reserved

시스템 컨트롤 명령어

Systemctl 을 통한 다양한 데몬/서비스 확인

- 서비스 상태 확인
  - systemctl status <servicename>.service
- 서비스 시작
  - systemctl start <servicename>.service
- 서비스 중지
  - systemctl stop <servicename>.service
- 서비스 재시작
  - systemctl restart <servicename>.service
- 설정 재로드
  - systemctl reload <servicename>.service
- 부팅시 서비스 자동 시작
  - systemctl enable <servicename>service
- 부팅시 서비스 자동 시작 삭제
  - systemctl disable <servicename>.service
- 서비스 숨기기 (시작불가)
  - systemctl mask <servicename>.service
- 서비스 숨기기 제거
  - systemctl unmask <servicename>.service

Copyright **FASTCAMPUS** Corp. All Rights Reserved

서비스 데몬의 로그 확인

Journalctl 을 통한 다양한 데몬/서비스 로그 확인

- 서비스 로그 확인
  - journalctl (전체로그)
  - journalctl -b (부팅후로그)
  - journalctl -f (최근로그 및 이후 로그 트래킹 대기)
- 특정 서비스의 로그 확인
  - journalctl -u <service-name>
- 특정 이벤트 속성 조회하기
  - journalctl -p crit (크리티컬 속성의 로그 확인)
- 특정 날짜로 조회하기
  - journalctl -u <service-name> --since=2020-06-01 --until=today

- 로그의 크기 확인
  - journalctl --disk-usage
- 로그 디렉토리 위치
  - /var/log/journal

로그레벨
- emerg (0)
- alert (1)
- crit (2)
- err (3)
- warning (4)
- notice (5)
- info (6)
- debug (7)

since / until
- 2020-05-01
- 2020-05-01 12:00:00
- yesterday, today, now

Copyright **FASTCAMPUS** Corp. All Rights Reserved

나만의 서비스 만들기

원하는 프로세스 서비스로 등록

- 기존 서비스 명령어의 설정파일 살펴보기
  - systemctl cat <service-name>.service
  - 예) systemctl cat sshd.service
    systemctl cat cups.service

- 서비스 명령파일 수정하기
  - systemctl edit --full <service-name>.service
  - 예) sudo systemctl edit --full sshd.service

- 새로운 데몬 서비스가 생성된 경우 그 라이브러리 목록 재 갱신
  - systemctl daemon-reload

Copyright **FASTCAMPUS** Corp. All Rights Reserved

나만의 서비스 만들기

원하는 프로세스 서비스로 등록

/usr/local/sbin/my-script.sh

```
#!/bin/bash
echo “I’m in $(date +%Y%m%d-%H%M%S)” >> /tmp/mylog.log
```

/etc/systemd/system/my-startup.service

```
[Unit]
Description=My Startup

[Service]
ExecStart=/usr/local/sbin/my-script.sh

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

```
$ systemctl status my-startup.service

# systemctl enable my-startup.service

# systemctl start my-startup.service

# chmod +x /usr/local/sbin/my-script.sh

# systemctl start my-startup.service

$ systemctl status my-startup.service
```

Fast campus
Copyright FASTCAMPUS Corp. All Rights Reserved

나만의 서비스 만들기

원하는 프로세스 서비스로 등록

/etc/systemd/system/my-daemon.service

```
[Unit]
Description=My First Daemon
After=network.target

[Service]
ExecStart=/usr/local/sbin/my-daemon.sh
Restart=always
RestartSec=5
User=user1
Group=user1
```

/usr/local/sbin/my-daemon.sh

```
#!/bin/bash
while true
do
   echo “I’m still in $(date +%Y%m%d-%H%M%S)”
   sleep 10
done
```

 # chmod +x my-daemon.sh

```
$ systemctl status my-daemon.service

# systemctl start my-daemon.service

$ journalctl -u my-daemon -f

# systemctl kill my-daemon.service
```

Fast campus
Copyright **FASTCAMPUS** Corp. All Rights Reserved