

Services & Settings

lctseng (2019-2020, CC BY-SA)
? (1996-2018)

交大資工系資訊中心

Computer Center of Department of Computer Science, NCTU

Handbook and Manual pages

- Official guide and be found at
 - Configuring Starting Services
 - <https://www.freebsd.org/doc/en/books/handbook/configtuning-starting-services.html>
 - https://www.freebsd.org/doc/zh_TW/books/handbook/configtuning-starting-services.html
 - Configuring RC
 - <https://www.freebsd.org/doc/en/books/handbook/configtuning-rcd.html>
 - https://www.freebsd.org/doc/zh_TW/books/handbook/configtuning-rcd.html
 - [rc\(8\)](#), [service\(8\)](#)

Basic Knowledge about Services

For FreeBSD

交大資工系資訊中心

Computer Center of Department of Computer Science, NCTU

The Service

- Background processes that provide services
 - A.k.a daemon process (usually with name xxx**d**)
 - Automatically started during boot
- Example
 - Standard services
 - NTP (ntpd)
 - Secure Shell (sshd)
 - Commonly known services
 - Web (httpd)
 - DNS (named)

Common Flow of Running a Service

1. Installation

- Through ports, packages, or source tarballs
 - `pkg install kde4`

2. Configuration

- Service specific configuration file(s)
- Make it start on boot
 - Manually update `/etc/rc.conf`
 - `kdm4_enable="YES"`
- Or using [service\(8\)](#) command
 - `service kdm4 enable`

Common Flow of Running a Service

3. Start the service now

- Calling script under rc.d/*
 - `/usr/local/etc/rc.d/kdm4 start`
- Or using "service"
 - `service kdm4 start`

4. Maintenance

- Updating, Restarting

Configuration Files

- /usr/local/etc
 - Configuration files of local installed programs are located there.

Daemon	User Program
/usr/local/etc/pure-ftpd.conf	/usr/local/etc/vim/vimrc
/usr/local/etc/apache24/httpd.conf	/usr/local/etc/screenrc

- Default config file usually installed with **.sample** suffix, **.default** suffix, or different suffix for different purpose. (**Copy and rename before using.**)
 - pure-ftpd.conf.sample
 - php.ini-dist
 - php.ini-recommended

Configuration Files

- A program with multiple config files are usually located in `/usr/local/etc/program-name/`
 - apache*
 - postfix
- Most configuration files have clear comment at the beginning or before each description

```
# pure-ftp.conf

# IP address/port to listen to (default=all IP and port 21)
Bind                                127.0.0.1,21

# Fork in background
Daemonize                            yes
```


Configuration Files

- Most popular styles
 - key <space> value
 - key = value
- Some with local effectiveness (e.g. http server)
 - Markup language-like:

```
<directory /path>  
    setting-for-this-path...  
</directory>
```

- Samba, rsync, devfs...

```
[xxxx]  
settings...  
[yyyy]  
settings...
```

```
<VirtualHost _default_:443>  
ServerAdmin lctseng@cs.nctu.edu.tw  
DocumentRoot "/usr/local/www/nic2015/"  
ServerName nic2015.nctu.cs.tw:443  
ErrorLog "/var/log/www.nic2015.error"  
CustomLog "/var/log/www.nic2015.common" common  
TransferLog "/var/log/www.nic2015.access"  
    <Directory "/usr/local/www/nic2015/">  
        AllowOverride All  
        Require all granted  
    </Directory>
```

```
[system=10]  
add path 'usb/*' mode 0660 group operator
```

RC Script

Scripts for starting / stopping a service

What does RC means?

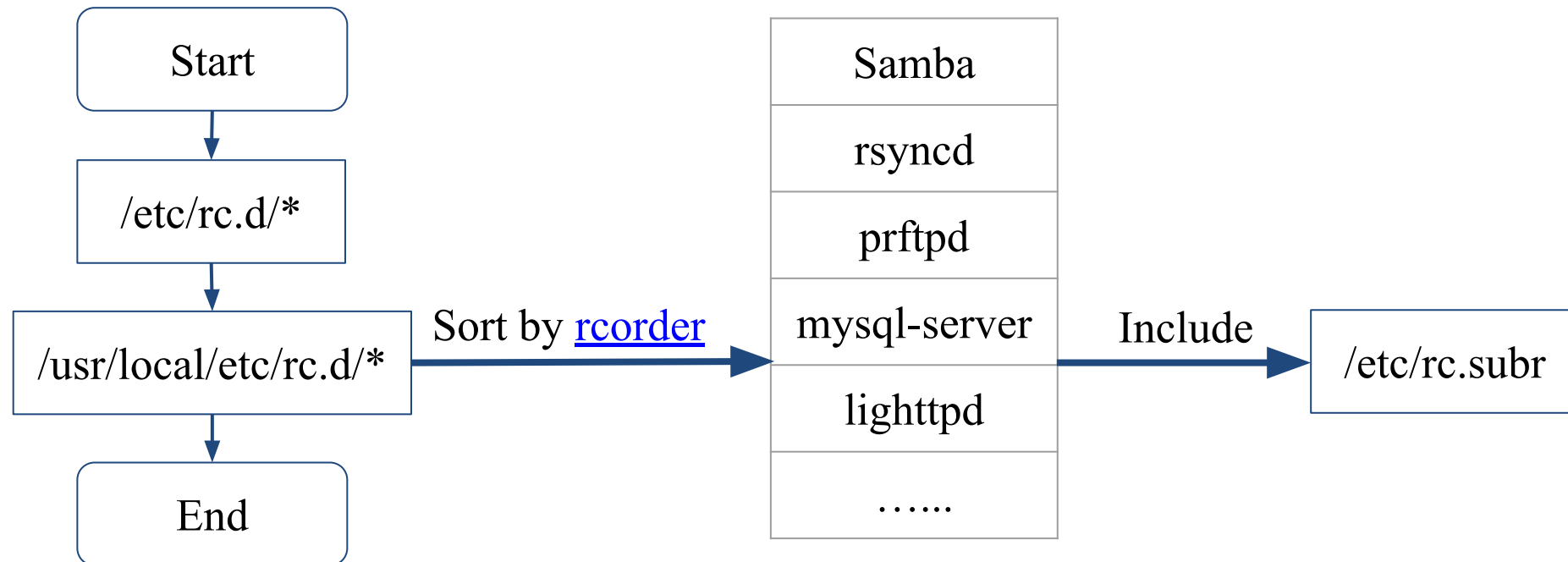
- **R**un **C**ommands (RunCom)
- Command scripts for auto-reboot and daemon startup
- [rc\(8\)](#)
- <https://www.freebsd.org/doc/handbook/configtuning-rcd.html>

Why do we need RC Script?

- Start services on system startup
- Starting and stopping services in a standard way
- Write configuration in the rc script for easy launch

with RC script	<pre>\$ service pure-ftpd start</pre>
without RC script	<pre>\$ /usr/local/sbin/pure-ftpd -g /var/run/pure-ftpd.pid -A -c50 -B -C8 -D -fftp -H -I15 -lpam -lunix -L10000:8 -m4 -s -U133:022 -u100 -k99 -Z</pre>

RC Script



- Dependency between each service is described in header of the script
- [rcorder\(8\)](#) is used to find out dependency ordering of each script
- Each rc script defines what to do when start / stop ...
- /etc/rc.subr defines what to do & check before / after start stop ...
 - [rc.subr\(8\)](#)

Components to launch daemon processes

- To launch a daemon process in background, we need:
- Launch command
 - Path to the executable binary/script
 - `/usr/sbin/inetd`
- Path to configuration file
 - Program-specified configuration (ports to use, files to read/write, ...)
 - `/etc/inetd.conf`
- Pidfile
 - Records (master) process id of the service
 - Other process (like “service” tool) can know what PID to show/kill
 - `/var/run/inetd.pid`

Inside the RC Script

- Example: /etc/rc.d/inetd

```
#!/bin/sh
#
# $FreeBSD: release/9.1.0/etc/rc.d/inetd 231653 2012-02-14
#
```

```
# PROVIDE: inetd
# REQUIRE: DAEMON LOGIN cleanvar
# KEYWORD: shutdown
```

```
. /etc/rc.subr
```

```
name="inetd"
rcvar="inetd_enable"
command="/usr/sbin/${name}"
pidfile="/var/run/${name}.pid"
required_files="/etc/${name}.conf"
extra_commands="reload"
```

```
load_rc_config $name
run_rc_command "$1"
```

for rcorder(8) to sort.

need to be included
by every RC script.

what to do with
start/stop/....

How to use rc script

- Example: ntpd

```
nctucs [~] -lctseng- /etc/rc.d/ntpd
Usage: /etc/rc.d/ntpd
[fast|force|one|quiet](start|stop|restart|rcvar|enable|disable|delete|enabled
|describe|extracommands|fetch|needfetch|status|poll
```

- After booting... (rc.conf)

```
nctucs [~] -lctseng- cat /var/db/ntp/ntpd.pid
4239
nctucs [~] -lctseng- ps aux | grep ntp
ntpd 4239  0.0  0.8 16488 16580  -  Ss   08:45      0:00.02 /usr/sbin/ntpd
-p /var/db/ntp/ntpd.pid -c /etc/ntp.conf -f /var/db/
```

- An easy way to access: "service"

- \$ service ntpd start/stop/restart/reload/...
 - Search /etc/rc.d and /usr/local/etc/rc.d

How to use rc script

- Start
 - Start the service and write PID files now
- Stop
 - Terminates the service by killing the process with PID recorded in pidfile
- Restart
 - Restart the service (or just start a new one if not running)
 - Some services implement 'restart' by 'stop + start'
- Enable/Disable
 - Edit /etc/rc.conf with `XXX_enable="YES"` or `"NO"`
 - Whether to automatically launch when booting

How to use rc script

- Status
 - Check the service is running or not
- Reload
 - Reload configuration file if the service support
- Rcvar
 - Show the variables used in rc.conf

```
nctucs [~] -lctseng- service pure-ftp rcvar
# pureftpd
#
pureftpd_enable="no"
# (default: "")
```

How to use rc script

- [one | fast | force]
 - One
 - Skip the check of rcvar="YES"
 - Start the service even if XXXX_enable="NO"
 - Force
 - Force start the service
 - Ignore any error it encountered (no prerequisite test)
 - ignore rcvar="YES" and set rc_force="YES"
 - Fast
 - Skip the check for an existing running process (pid check)
 - Set rc_fast="YES"



Local installed service

- More about how to use rc.conf for an installed service, read comments from that script
 - /usr/local/etc/rc.d/pure-ftpd

```
# Add the following lines to /etc/rc.conf to enable pure-ftpd:
#
# pureftpd_enable="yes"
# pureftpd_flags("<set as needed>")
#
# Add the following lines to /etc/rc.conf to enable pure-authd daemon:
#
# pureftpd_authd_enable="yes"
# pureftpd_authdscript="/full/path/to/auth_script"
# pureftpd_authsocket="/var/run/ftpd.sock"
#
# Add the following lines to /etc/rc.conf to enable uploadscript daemon:
#
# pureftpd_upload_enable="yes"
# pureftpd_uploadscript="/full/path/to/upload_script"
```

System-V

Startup Scripts

- SystemV-style startup scripts
 - Sun
 - /etc/init.d/ 
 - /etc/rc.d/rcn.d/  Symbolic link
 - Each script is responsible for one daemon or one aspect of system.

Example: sshd in SunOS

```
case "$1" in
'start')
    if [ -x /usr/local/sbin/sshd ]; then
        echo "Starting the secure shell daemon"
        /usr/local/sbin/sshd &
    fi
;;
'stop')
    echo "Stopping the secure shell daemon "
    pkill -TERM sshd
;;
*)
    echo "Usage: /etc/init.d/sshd { start | stop }"
;;
esac
exit 0
```

Startup Scripts – SystemV-style startup scripts (1)

- /etc/rc.d/rcn.d/
 - When init transitions from lower run level to higher one,
 - It runs all the scripts that start with "S" in ascending order with "start" argument
 - When init transitions from high run level to lower one,
 - It runs all the scripts that start with "K" in descending order with "stop" argument

```
linux4 fyli ... > rc.d cd /etc/rc.d/
linux4 fyli ... > rc.d ls
init.d rc0.d rc1.d rc2.d rc3.d rc4.d rc5.d rc6.d rc.local
linux4 fyli ... > rc.d cd rc0.d/
linux4 fyli ... > rc0.d ls
K50netconsole K90network
linux4 fyli ... > rc0.d
```

Startup Scripts –

SystemV-style startup scripts (2)

- If you write a daemon and want init to start/stop it,
 - write a script and put in /etc/init.d
 - make suitable symbolic link in rcn.d
 - `ln -s /etc/init.d/initiald /etc/rc2.d/S61initiald`
 - `ln -s /etc/init.d/initiald /etc/rc0.d/K33initiald`

Systemd

Systemd

- Evolved from System-V
 - Backward compatibility
- Goal: provide a faster booting process
 - Less processes
 - Parallel launching
- Use "targets" replace run-levels
 - Easier to use
- Tool
 - [systemctl \(1\)](#)
 - Similar to "service" tool in FreeBSD

Flow of Running a Service with Systemd

Example with Ubuntu

- Installation
 - Through packages, or source tarballs
 - `apt install apache2`
- Configuration
 - Service specific configuration file(s)
 - `/etc/apache2/*`
 - `systemctl enable apache2`
- Start
 - `systemctl start apache2`
- Maintenance
 - Updating, Restarting

How to use systemctl

- Usage
 - `systemctl [OPTIONS...] {COMMAND} ...`
- Common commands
 - `enable / disable`
 - Enable/disable launch when booting
 - `start / stop / reload / restart / status`
 - `condrestart`
 - Restart only if service is running

Unit files

- Define services
 - `/lib/systemd/system/*`
 - Similar to `/etc/rc.d/*` in FreeBSD
- `systemd.service` (5)
- Example 1: (simple service)
 - Type
 - Simple: main process keeps running
 - Forking: main process forks and exits
 - ExecStart
 - Command to launch the service
 - WantedBy
 - Run this service at which target

```
[Unit]
Description=Some simple daemon

[Service]
Type=forking
ExecStart=/usr/sbin/my-simple-daemon -d
PIDFile=/var/run/my-daemon.pid

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

Unit files

- apache2.service
 - After
 - Dependency. Start service after dependency is fulfilled
 - ExecStop / ExecReload
 - Custom command to stop / reload the service

```
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target

[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl stop
ExecReload=/usr/sbin/apachectl graceful
PrivateTmp=true
Restart=on-abort

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

Unit files

- When enable a service, it will create links from "lib/systemd/system/*.service" to "etc/systemd/system/XXX.target.wants/*"
- /etc/systemd/system/multi-user.target.wants/apache2.service
-> /lib/systemd/system/apache2.service

```
13:18 lctseng@lctseng-sa-ubuntu(10.0.2.15) [/etc/systemd/system/multi-user.target.wants]
[XD] % ll
total 8
drwxr-xr-x  2 root root 4096 Sep 30 12:21 .
drwxr-xr-x 15 root root 4096 Sep 28 23:25 ..
lrwxrwxrwx  1 root root   35 Sep 30 12:21 apache2.service -> /lib/systemd/system/apache2
lrwxrwxrwx  1 root root   31 Aug  5 19:24 atd.service -> /lib/systemd/system/atd.service
lrwxrwxrwx  1 root root   41 Aug  5 19:23 console-setup.service -> /lib/systemd/system/c
lrwxrwxrwx  1 root root   32 Aug  5 19:23 cron.service -> /lib/systemd/system/cron.servi
lrwxrwxrwx  1 root root   36 Aug  5 19:24 ebttables.service -> /lib/systemd/system/ebtabl
lrwxrwxrwx  1 root root   38 Aug  5 19:24 irqbalance.service -> /lib/systemd/system/irqb
lrwxrwxrwx  1 root root   33 Aug  5 19:24 lxcfs.service -> /lib/systemd/system/lxcfs.ser
lrwxrwxrwx  1 root root   42 Aug  5 19:24 lxd-containers.service -> /lib/systemd/system/
```



Appendix: Other init systems

Different Init Systems

	BSD rc	sysvinit (System V)	systemd	launchd	OpenRC
Operating Systems	NetBSD, FreeBSD, OpenBSD	Debian (< 8.0)	Ubuntu, Debian (≥ 8.0)	Mac OS X	Gentoo Linux, FreeBSD, NetBSD
Example commands	service	init, telinit, runlevel	systemctl	launchctl	rc-service, rc-update
Service config files	/etc/rc.d	/etc/init.d/	/lib/systemd/ system/	/System/Library/LaunchD aemons	/etc/init.d/ 