



systemd

chenshh

Before we start

systemd is only for linux

ONLY FOR LINUX

NO UNIX

History

1. BSD Style init
2. sysvinit
3. upstart
4. systemd

BSD Style Init

1./etc/rc.conf

2./etc/rc.d/ /usr/local/etc/rc.d

Shell Script

Keep It Simple Stupid

Very Old BSD init (PDP-7)

`/etc/rc`

Take a look at startup sequence

/sbin/init

/bin/sh

/etc/rc -> /etc/rc.conf , /etc/rc.conf.d

/etc/netstart (Not used at startup in freebsd , but for manually start network)

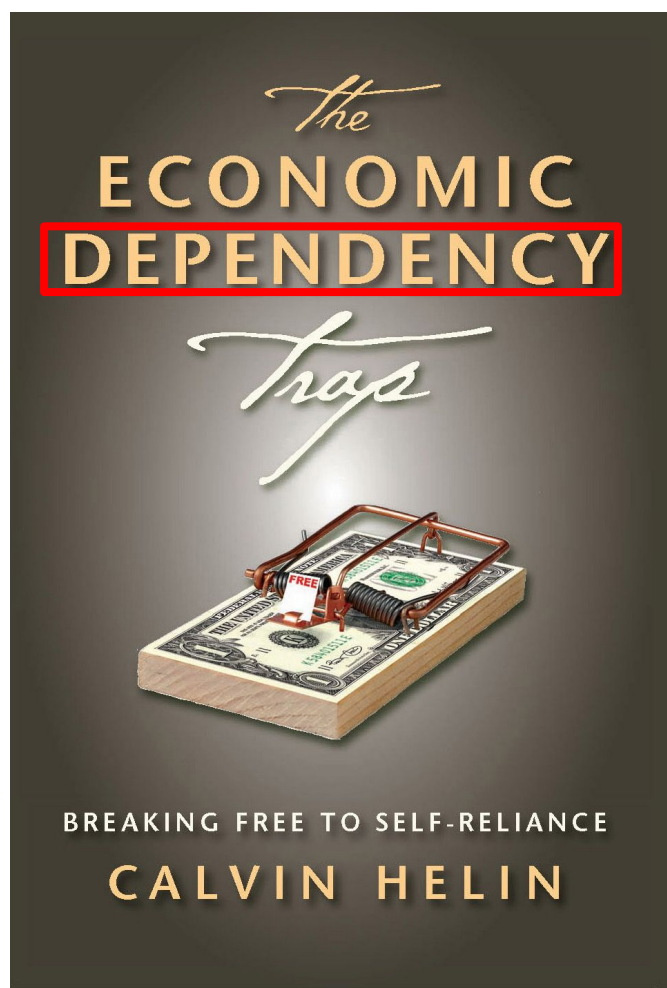
/etc/rc.d,/usr/local/rc.d

/etc/rc.local

init for itself , run getty

Old rc (BSD 4.4) vs New rc (NetBSD 1.5)

只是這張圖潮潮der

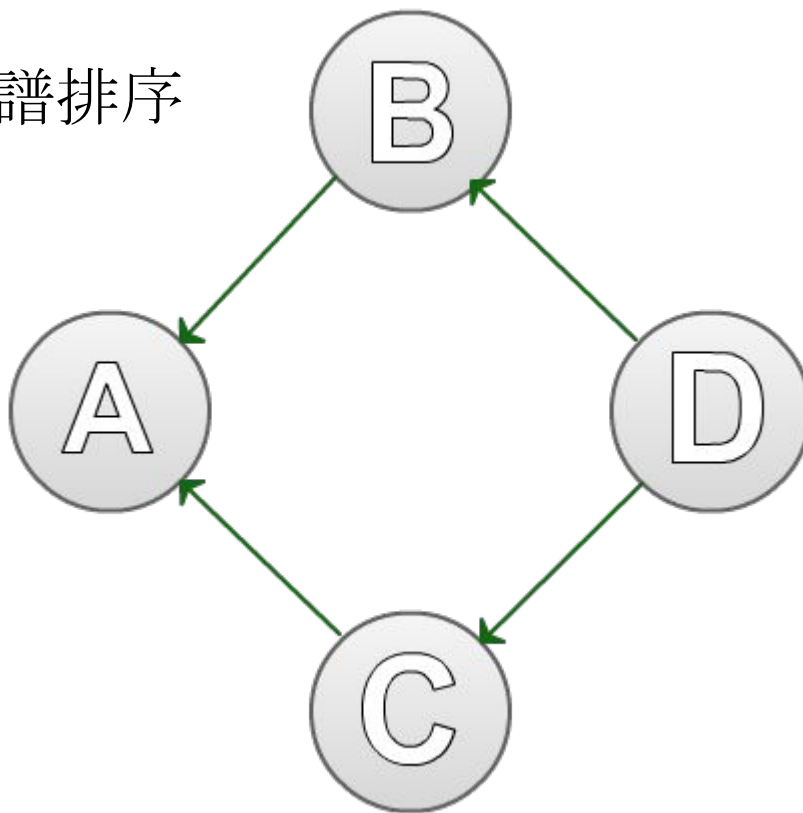


What is Dependency?

A **D**irected **A**cyclic **G**raph

有向無環圖

必存在一拓譜排序



Old way

by rc.conf period.

Human will make mistakes.

New Way

Using /sbin/rcorder to provide dependency

```
#!/bin/sh
# PROVIDE: mumbled oldmumble
# REQUIRE: DAEMON cleanvar frotz2
# BEFORE: LOGIN3
# KEYWORD: nojail shutdown4
```

sysvinit

1. First developed by **System V** R4 (AT&T)
2. SHELL SCRIPT
3. define runlevels
4. init 3 init 5

runlevels

0

Shut down the operating system so that it's safe to turn off the power.

s or S

Single user mode, with all file systems mounted.

1

Single user mode, with all file systems mounted and user logins allowed.

2

Multi user mode, with all services running except NFS server daemons.

3

Multi-user mode with all services running. This is usually the default.

4

Currently unused.

5

Shut down the system and attempt to turn off the power.(Differ in distros , the newer mostly means X)

6

Shut down the system to level 0, and reboot.

Take a look at startup sequence

/sbin/init

/etc/rc.d/rc.sysinit -> Runlevel N (System bootup)

read /etc/inittab

Runlevel S -> /etc/rcS.d

Runlevel X -> /etc/rcX.d

rcX.d?

To kind of names

SXXXXXXXXXXXXXXXXX -> Startup

KXXXXXXXXXXXXXXXXX -> Kill

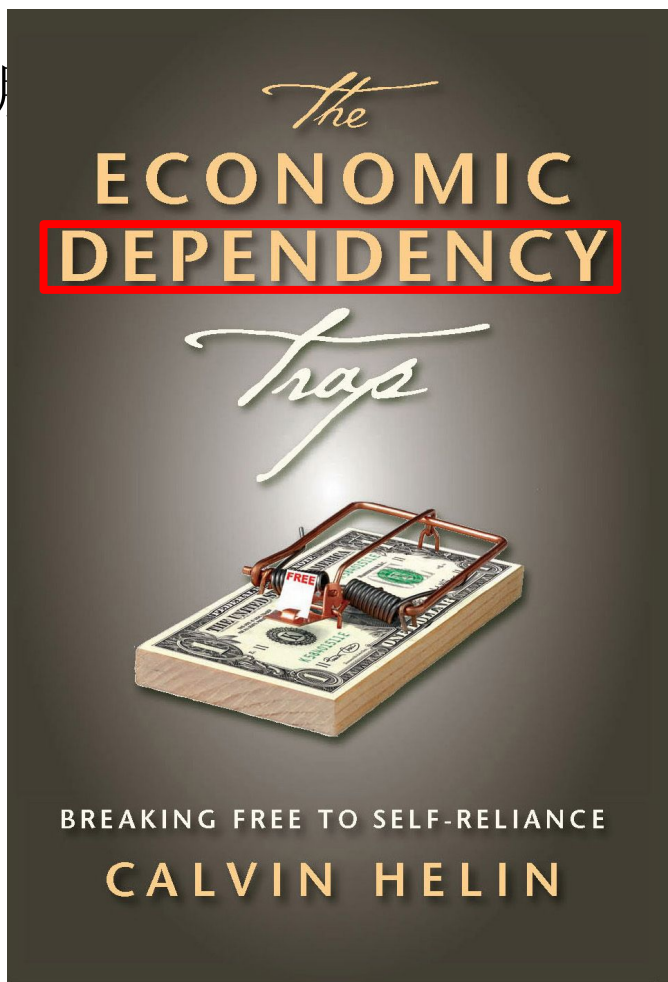
Run in lexicographical

/etc/inittab

```
1 ap::sysinit:/sbin/autopush -f /etc/iu.ap
2 ap::sysinit:/sbin/soconfig -f /etc/sock2path
3 fs::sysinit:/sbin/rcS sysinit >/dev/msglog 2<>/dev/msglog </dev/console
4 is:3:inittdefault:
5 p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...
6 sS:s:wait:/sbin/rcS >/dev/msglog 2<>/dev/msglog </dev/console
7 s0:0:wait:/sbin/rc0 >/dev/msglog 2<>/dev/msglog </dev/console
8 s1:1:respawn:/sbin/rc1 >/dev/msglog 2<>/dev/msglog </dev/console
9 s2:23:wait:/sbin/rc2 >/dev/msglog 2<>/dev/msglog </dev/console
10 s3:3:wait:/sbin/rc3 >/dev/msglog 2<>/dev/msglog </dev/console
11 s5:5:wait:/sbin/rc5 >/dev/msglog 2<>/dev/msglog </dev/console
12 s6:6:wait:/sbin/rc6 >/dev/msglog 2<>/dev/msglog </dev/console
13 fw:0:wait:/sbin/uadmin 2 0 >/dev/msglog 2<>/dev/msglog </dev/console
14 of:5:wait:/sbin/uadmin 2 6 >/dev/msglog 2<>/dev/msglog </dev/console
15 rb:6:wait:/sbin/uadmin 2 1 >/dev/msglog 2<>/dev/msglog </dev/console
16 sc:234:respawn:/usr/lib/saf/sac -t 300
17 co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
    -T terminal-type -d /dev/console -l console -m ldterm,ttcompat
```

How to provide dependency?

這張圖真的潮



So simple

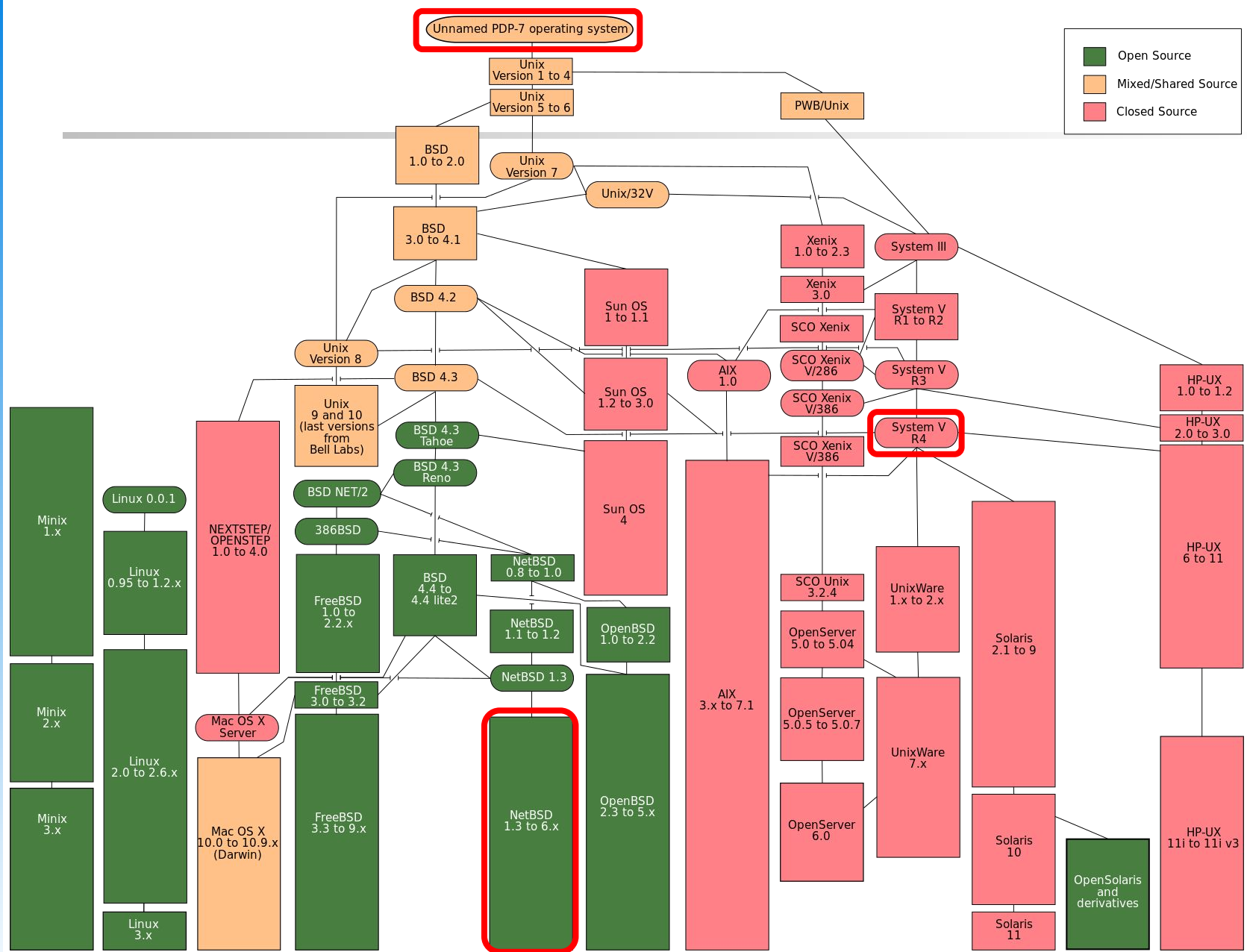
S10-abc

S11-foo

S22-bar

Easy to debug!

Lets talk about unix history



Time for New Generations!

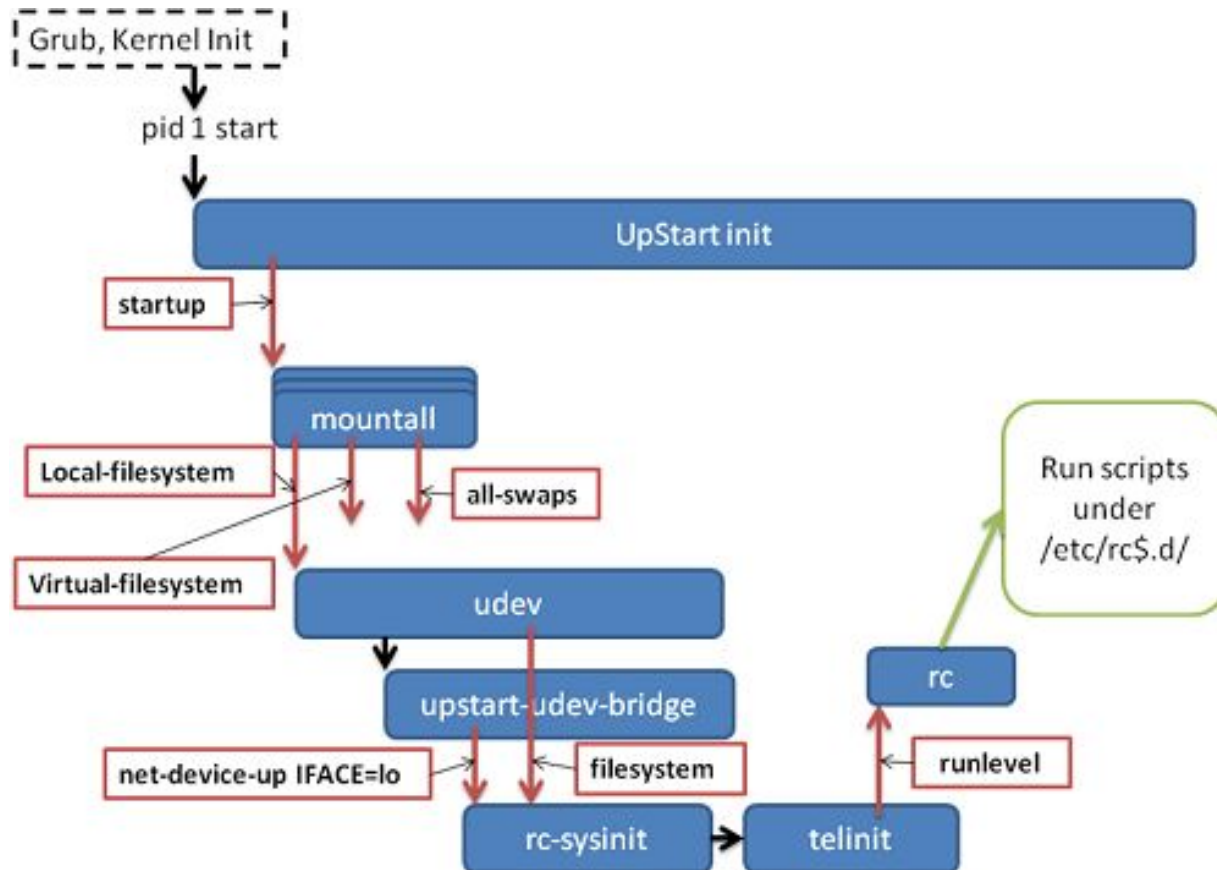
Why replacing Old style of init

1. Linux on desktop, which shutting down and booting is more often.
2. Need more userfriendly way to add services.
3. Shell script is slow.
4. Services can't be started parallelly.
5. Some services is on-demand. ex. CUPS

upstart

1. Ubuntu
2. Triggered by event
3. Can start/kill services by event
4. using udev bridge to implement hotplug events

Taka a look at startup sequence



Jobs and Events

Jobs:

Task Job

Service Job

Abstract Job

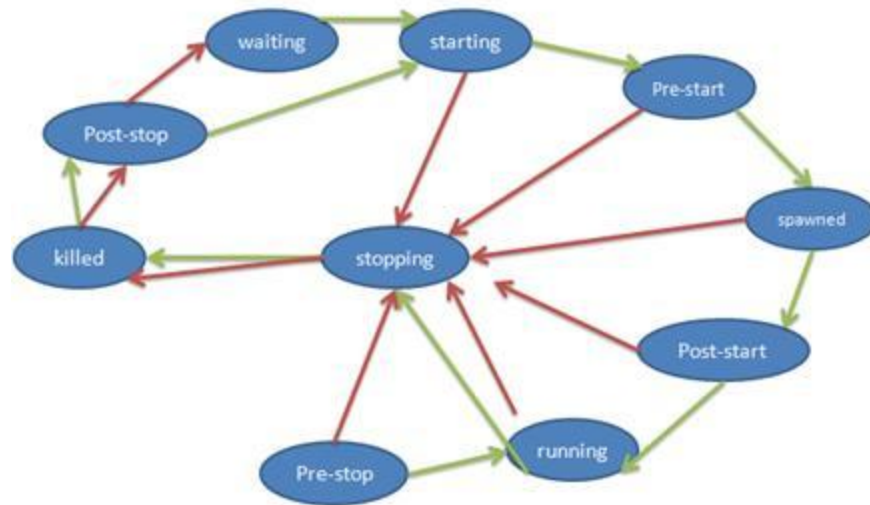
Events:

Signals

Method

Hooks

Job Life Cycle



upstart config file

```
start on runlevel [2345]
stop on runlevel [!2345]

# start on eventname

task

# XXX: configurable
export user
script
uid=$(getent passwd "$user"|cut -d: -f3)
gid=$(getent passwd "$user"|cut -d: -f4)

# Create directory that would normally be
# created by PAM when a user logs in.
export XDG_RUNTIME_DIR="/run/user/$uid"
mkdir -p "$XDG_RUNTIME_DIR"
chmod 0700 "$XDG_RUNTIME_DIR"
chown "$uid:$gid" "$XDG_RUNTIME_DIR"
start session_init USER="$user"
end script
```

systemd

cgroup

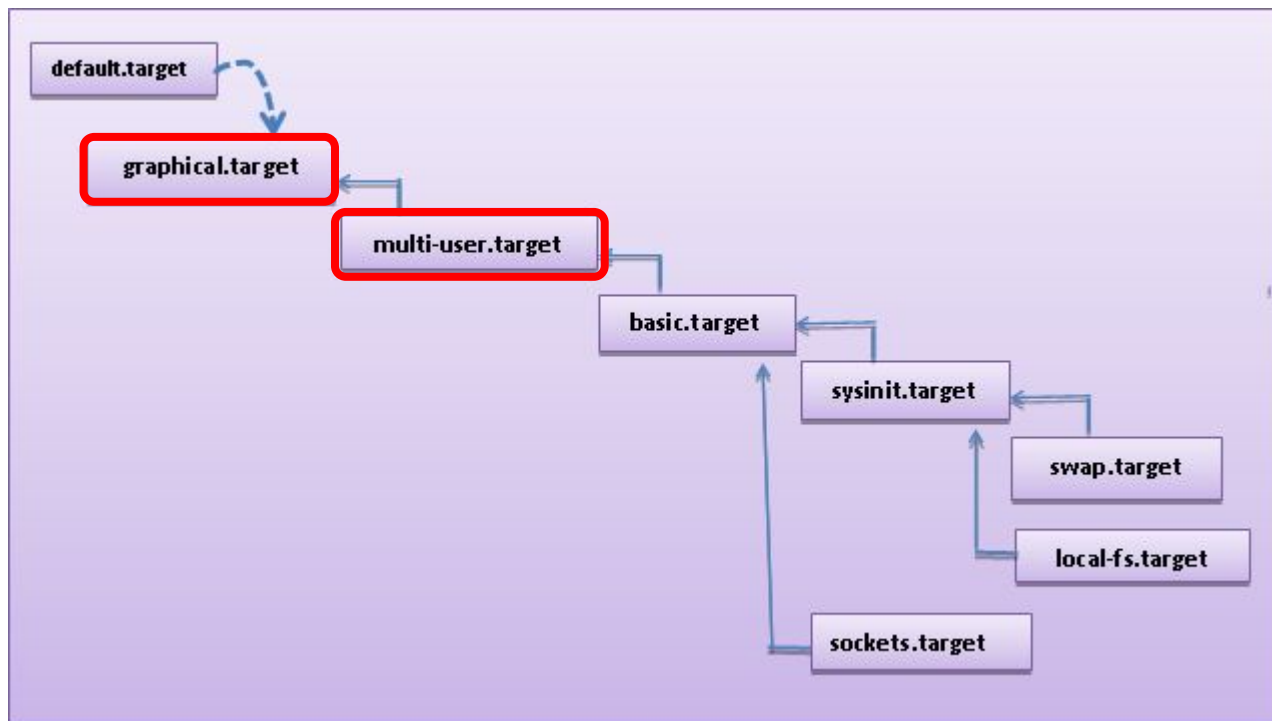
dependency

sockets

journal

lots of things

Take a look at startup sequence



Targets , Units(Services , Sockets ...)

Targets Wants Units

Sockets Triggers Services

Services

[Unit]

```
Description=MyApp  
After=docker.service  
Requires=docker.service
```

[Service]

```
TimeoutStartSec=0  
ExecStartPre=/usr/bin/docker kill busybox1  
ExecStartPre=/usr/bin/docker rm busybox1  
ExecStartPre=/usr/bin/docker pull busybox  
ExecStart=/usr/bin/docker run --name busybox1 busybox /bin/sh -c "while true; do echo Hello  
World; sleep 1; done"
```

[Install]

```
WantedBy=multi-user.target
```

Dependency

Require & After/Before

Wants

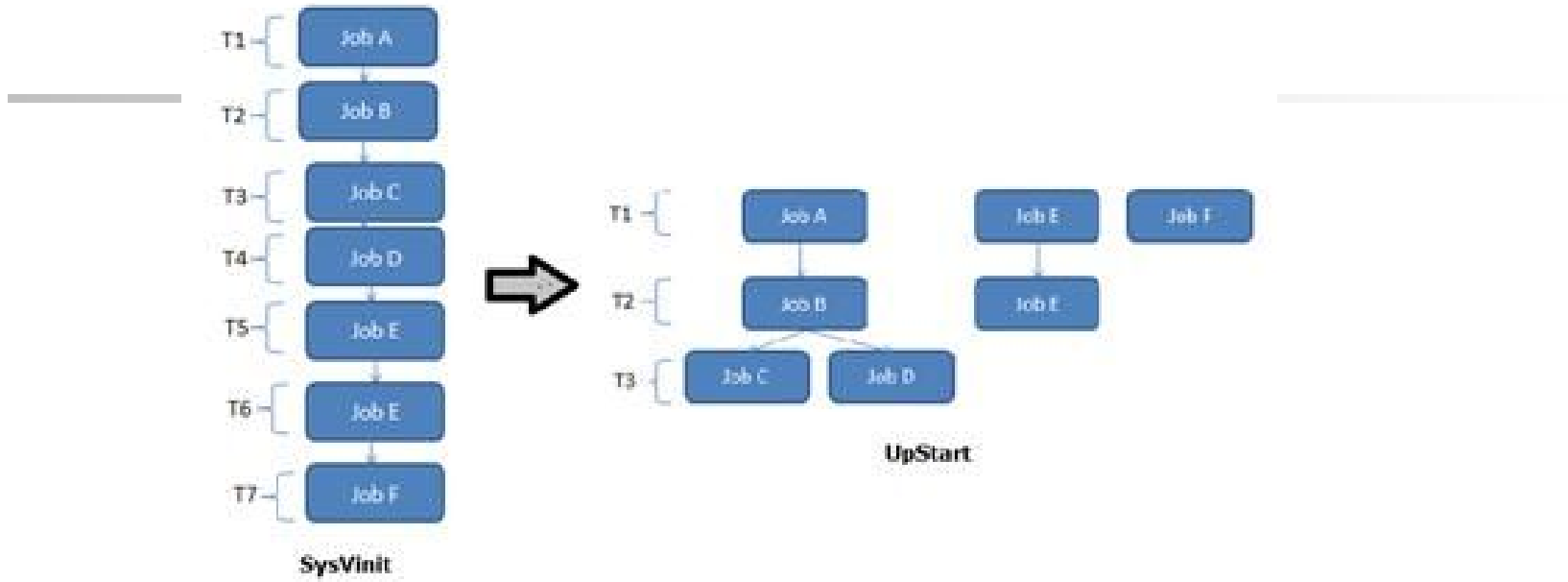
```
[Unit]
Description=I want B
Wants=B.service
...
```

```
[Unit]
Description=I Require B start Before me
Requires=B.service
After=B.sevice
...
```

Why Sockets?

On-demand services

Improve startup speed



Forking can derive socket fds
ex.inetd , xinetd

Sockets

```
[Unit]
  Description=SSH Socket for Per-
  Connection Servers

[Socket]
  ListenStream=22
  Accept=yes

[Install]
  WantedBy=sockets.target
```

```
[Unit]
  Description=SSH Per-Connection Server

[Service]
  ExecStart=-/usr/sbin/sshd -i
  StandardInput=socket
```

PIDs

Traditional

PID files

Upstart

Need to define how many forks
will the process do

Systemd

cgroup

cgroup

implemented since linux kernel 2.6.24

used to control resources of processes

You can control it by editing `/sys/fs/cgroup`

journald

replace syslog

syslog is too simple!

You can say you are any PID

No consistent format

We can use unit files to control logs

automount

replace autofs

networkd

replace network scripts

And so many other features

machined - virtual machines

hostnamed - set hostname

timedated - timezone , ntp

localed - languages

logind - user session management

importd - image management

resolved - DNS settings

Reference

<http://goo.gl/fu925Q>