Userspace Initialization

Advanced Operating Systems and Virtualization

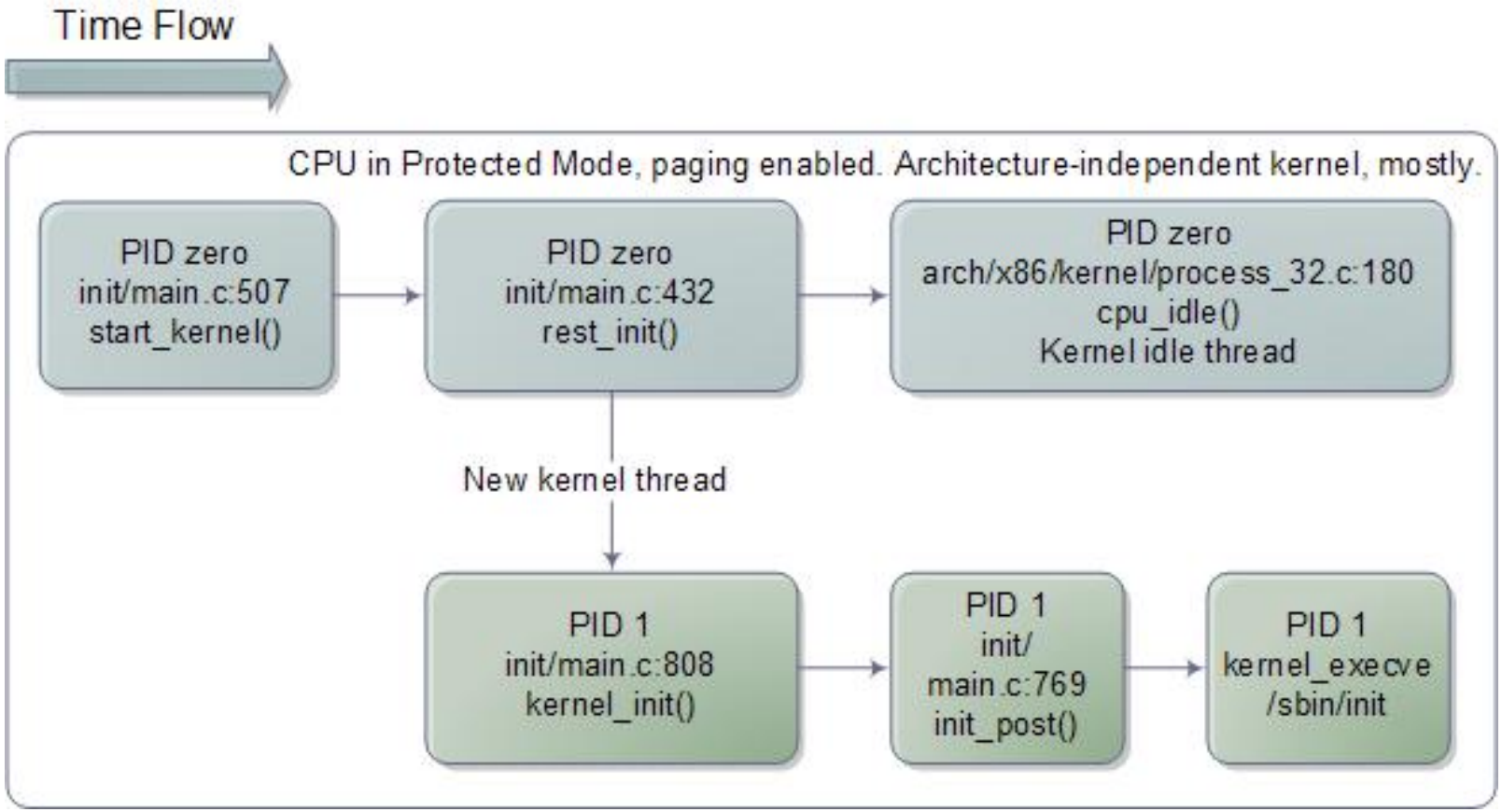
Alessandro Pellegrini

A.Y. 2019/2020

Boot Sequence

BIOS/UEFI	The actual Hardware Startup
Bootloader Stage 1	Executes the Stage 2 bootloader (skipped in case of UEFI)
Bootloader Stage 2	Loads and starts the Kernel
Kernel Startup	The Kernel takes control of and initializes the machine (machine-dependent operations)
Init	**First process: basic environment initialization (e.g., SystemV Init, systemd)**
Runlevels/Targets	Initializes the user environment (e.g., single-user mode, multiuser, graphical, ...)

Back to Kernel Initialization



`rest_init()`

- We have to “leave” the infinite loop in pid 0
  - We need to start other processes than idle!
- A new **kernel thread** is created, referencing `kernel_init()` as its entry point
- A call to `schedule()` is issued, to start scheduling the newly-created process
- This is done right before PID 0 calls into `cpu_idle()`

Starting `/sbin/init`

- `/sbin/init` is the first userspace process ever started
- This process is commonly stored into the ramdisk, to speedup the booting process
- init will have to load configuration files from the hard drive
- This means that the VFS, Device Management, and Interrupt subsystems must be initialized *before* loading init

Boot Sequence

BIOS/UEFI	The actual Hardware Startup
Bootloader Stage 1	Executes the Stage 2 bootloader (skipped in case of UEFI)
Bootloader Stage 2	Loads and starts the Kernel
Kernel Startup	The Kernel takes control of and initializes the machine (machine-dependent operations)
Init	First process: basic environment initialization (e.g., SystemV Init, systemd)
Runlevels/Targets	Initializes the user environment (e.g., single-user mode, multiuser, graphical, ...)

Startup Services

- Hostname
- Timezone
- Check the hard drives
- Mount the hard drives
- Remove files from /tmp
- Configure network interfaces
- Start daemons and network services

Startup Run Levels

Level	Mode
1 (S)	Single user
2	Multiuser (no networking)
3	Full Multiuser
4	Unused
5	X11
6	Reboot
0	Halt

Run Level Scripts

- Actual scripts placed in: `/etc/rc.d/init.d/`
- `/etc/rc.d/rc#.d/`:
  - Symbolic links to `/etc/init.d` scripts
  - S## - Start scripts
  - K## - Stop scripts
  - `/etc/sysconfig/`: script configuration files

- `chkconfig <script> on|off`
- `service <script> start|stop|restart`

/etc/inittab

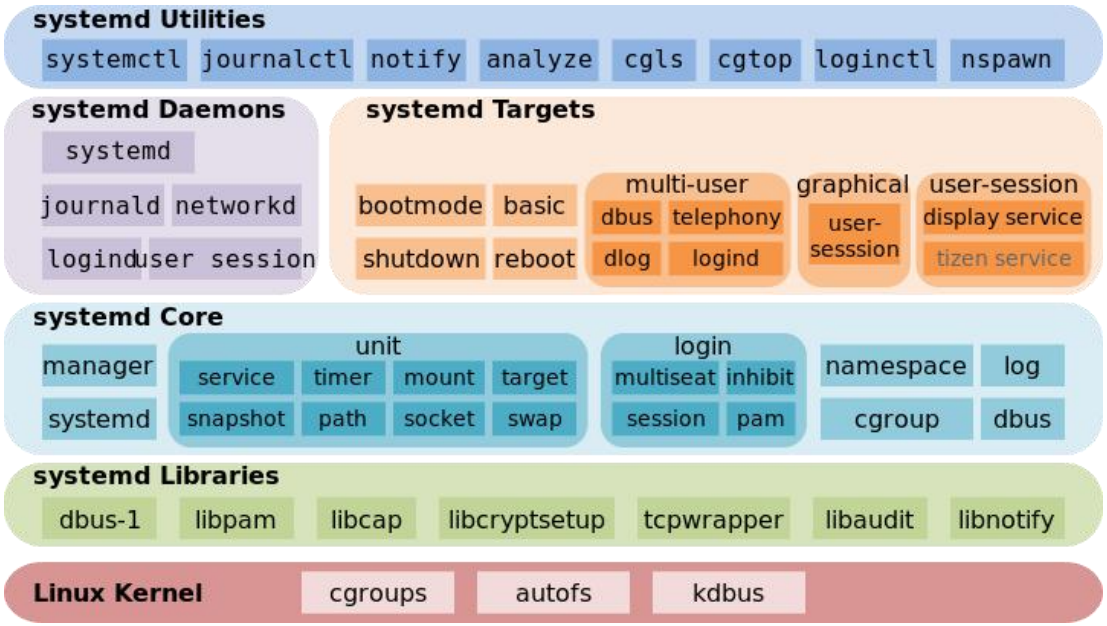
- Initializes system for use
- Format: `id:rl:action:process`
  - `id`: uniquely identifies entry
  - `rl`: what runlevels the entry applies to
  - `action`: the type of action to execute
  - `process`: process command line

- An example:

```
2:23:respawn:/sbin/getty 38400 tty2
```

Systemd

- Becoming more prevalent in Linux Distros
- Mostly compatible with the init system
  - init scripts could be read as alternative format
- Based on the notion of "units" and "dependencies"



Systemd Targets

- The concept of "runlevel" is mapped to "targets" in systemd jargon
- Runlevel is defined through a symbolic to one of the runlevel targets
- Runlevel Target
  - Runlevel 3:
    /lib/systemd/system/multi-user.target
  - Runlevel 5:
    /lib/systemd/system/graphical.target
- Change Runlevel:
  - Remove current link `/etc/systemd/system/default.target`
  - Add a new link to the desired runlevel

Systemd Unit Types

- Different unit types control different aspects of the operating system
  - service: handles daemons
  - socket: handles network sockets
  - target: logical grouping of units (example: runlevel)
  - device: expose kernel devices
  - mount: controls mount points of the files system
  - automount: mounts the file system
  - snapshot: references other units (similar to targets)

Systemd Unit Section

- [Unit]
  - Description: A meaningful description of the unit
  - Requires: Configures dependencies on other units
  - Wants: Configures weaker dependencies
  - Conflicts: Negative dependencies
  - Before: This unit must be started before these others
  - After: This unit must be started after these others (unlike Requires, it doest not start the unit if not already active)

Systemd Service Section

- [Service]
  - Type= simple|oneshot|forking|dbus|notify|idle
  - ExecStart
  - ExecReload
  - ExecStop
  - Restart=no|on-success|on-failure|on-abort|always

Systemd Install Section

- [Install]
  - Wantedby=

- Used to determine when to start
  (e.g. Runlevel)

An Example

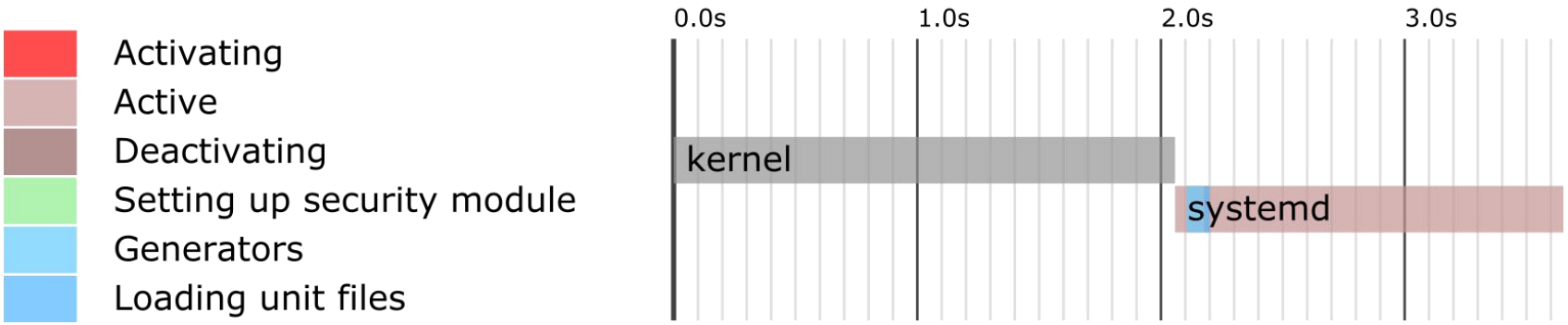
```
[Unit]
Description=Postfix Mail Transport Agent
After=syslog.target network.target
Conflicts=sendmail.service exim.service

[Service]
Type=forking
PIDFile=/var/spool/postfix/pid/master.pid
EnvironmentFile=-/etc/sysconfig/network
ExecStartPre=-/usr/libexec/postfix/aliasesdb
ExecStartPre=-/usr/libexec/postfix/chroot-update
ExecStart=/usr/sbin/postfix start
ExecReload=/usr/sbin/postfix reload
ExecStop=/usr/sbin/postfix stop

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

Boot Sequence

BIOS/UEFI	The actual Hardware Startup
Bootloader Stage 1	Executes the Stage 2 bootloader (skipped in case of UEFI)
Bootloader Stage 2	Loads and starts the Kernel
Kernel Startup	The Kernel takes control of and initializes the machine (machine-dependent operations)
Init	First process: basic environment initialization (e.g., SystemV Init, systemd)
Runlevels/Targets	Initializes the user environment (e.g., single-user mode, multiuser, graphical, ...)

Boot Sequence

Startup finished in 2.057s (kernel) + 1.593s (userspace) = 3.650s



Times taken on a 4-core Intel i7-7600U CPU @ 2.80GHz (7th gen) with 16 GB of RAM