

Mumbai Technology Meetup, NaviMumbai
December 14, 2014

Linux Kernel & building of Linux Kernel from scratch

Siji Sunny

History

1991 - Finnish student Linus Torvalds to create a new free operating system kernel named as Linux

Started as hobby project - August 1991, released Linux under its own license

1992 – released under GNU GPL license

Linux and GNU developers worked to integrate GNU components with Linux to make a fully functional and free operating system

1993: Over 100 developers work on the Linux kernel

Present

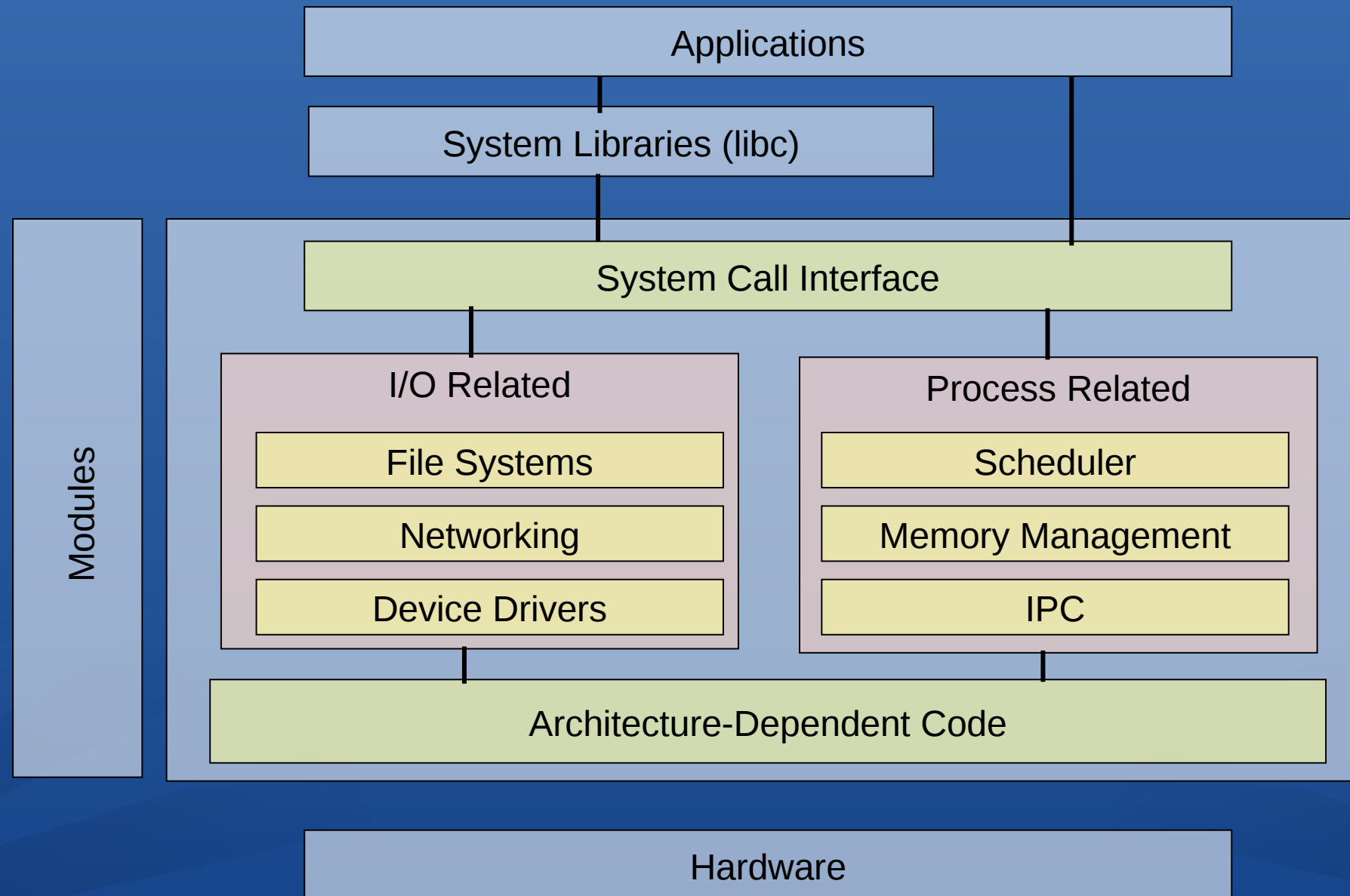
Supports Multiple architectures (ARM, X86, MIPS, SPARC, etc)

Google's Linux-based Android claims 75% of the smart-phone market share, in terms of the number of phones shipped

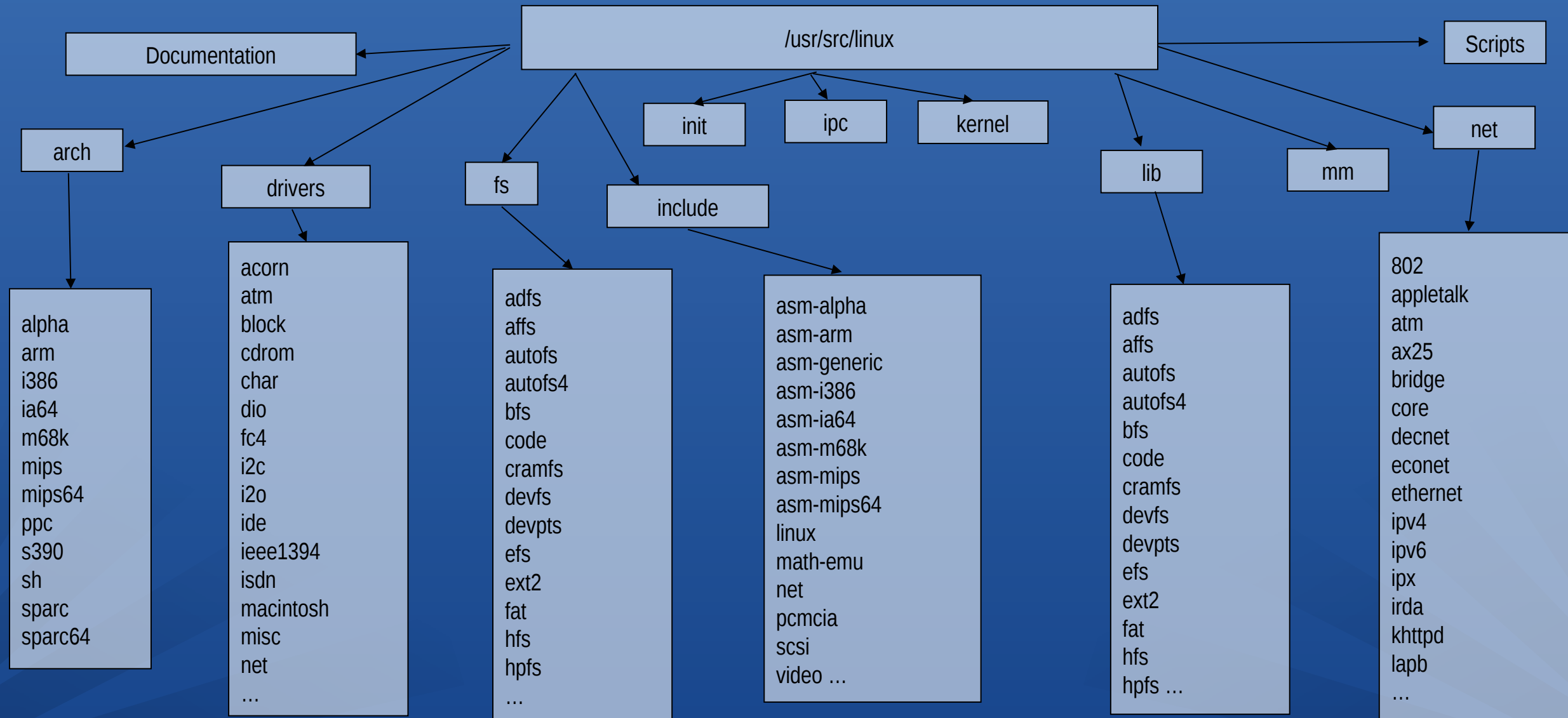
Ubuntu claims 22,000,000 users

3.18 version in 2014 with more than 16 million lines of source code

Architecture



Kernel Source Tree



Linux-3.18/..

- linux-3.18/arch
 - *Subdirectories for each current port.*
 - ***arch*** as supports :
alpha, arm, i386, ia64, m68k, mips, mips64, ppc, s390, sh, sparc, sparc64.
- linux-3.18/drivers
- linux-3.18/fs
 - *virtual filesystem (VFS) framework.*
 - *subdirectories for actual filesystems.*
- linux-3.18/include

Linux-3.18/..

- linux-3.18/include
 - include/asm-*
Architecture-dependent include subdirectories.
 - include/linux:
Header info needed both by the kernel and user apps. (linked to /usr/include/linux.)
- linux-3.18/init
 - contains the version banner that prints at boot.
 - architecture-independent boot code.
- linux-3.18/kernel – The core kernel code

Linux-3.18/..

- linux-3.18/kernel
 - *The core kernel code*
 - *sched.c – “the main kernel file”: scheduler, wait queues, timers, alarms, task queues.*
 - *Process control: - fork.c, exec.c, signal.c, exit.c etc...*
 - *Kernel module support:- kmod.c, ksyms.c, module.c.*
- linux-3.18/scripts
 - *Menu-based kernel configuration.*
 - *Kernel patching.*
 - *Generating kernel documentation*

Compile & Build From Source

To enable New/ experimental features that are not part of the default kernel.

To enable support for a new hardware (Drivers) that is not currently supported by the default kernel.

To debug the kernel

To Improve the performance

Get latest Kernel source

```
# wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.18.tar.xz
```

Or Download directly & Extract -

```
https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.18.tar.xz
```

Untar the Kernel Source -tar -xvJf linux-3.18.tar.xz

Configure Kernel

- The kernel contains nearly 3000 configuration options
- Generally Includes supports for most of the hardwares
- Starts with the basic configuration provided by the vendor or kernel

```
# cd linux-3.9.3
```

```
# make menuconfig
```

make config - starts a character based questions and answer session

make menuconfig - starts a terminal-oriented configuration tool (using ncurses)

make xconfig - starts a X based configuration tool

make menuconfig

File Edit View Search Terminal Help

.config - Linux/x86 3.18.0 Kernel Configuration

Linux/x86 3.18.0 Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
[ ] 64-bit kernel
    General setup --->
    [*] Enable loadable module support --->
    [*] Enable the block layer --->
        Processor type and features --->
        Power management and ACPI options --->
        Bus options (PCI etc.) --->
        Executable file formats / Emulations --->
    [*] Networking support --->
        Device Drivers --->
        Firmware Drivers --->
        File systems --->
        Kernel hacking --->
        Security options --->
    *- Cryptographic API --->
    *- Virtualization --->
        Library routines --->
```

<Select>

< Exit >

< Help >

< Save >

< Load >

Configure .config file

- The .config file contains everything that is specific to the kernel
- An option will either indicate some driver is built into the kernel ("=y") or will be built as a module ("=m") or is not selected.
- The unselected state can either be indicated by a line starting with "#" or by the absence of the relevant line from the .config file.
- Keep old kernel settings – make oldconfig

Compile the Linux Kernel

Compile the main kernel

```
# make
```

*(make -j N" with N the number of (core * thread-by-core) to help built faster)*

Compile the kernel modules

```
# make modules
```

(Create a directory named /lib/modules/3.9.3/ in your system)

Install the kernel modules

```
# make modules_install
```

Install the New Kernel

```
# make install
```

This creates following files in the /boot directory

- vmlinuz-3.18 – The actual kernel
- System.map-3.18 – The symbols exported by the kernel
- initrd.img-3.18 – initrd image is temporary root file system used during boot process
- config-3.18 – The kernel configuration file

Updates grub – Add new kernel boot entry

```
# reboot
```

Kernel Modules

- Drivers can be loaded as modules – Dynamically and statically
- Drivers loaded at boot time (part of linux kernel image) – Static drivers
- Dynamic modules can be loaded after kernel bootup, using insmod / modprobe
 - Insmod – insert module into a kernel

Questions

Siji Sunny
sijisunny@gmail.com

Twitter - siji_sunny

Blog : www.sijisunny.com