

ADMINISTRASI SISTEM

STT TERPADU NURUL FIKRI
TEKNIK INFORMATIKA & SISTEM INFORMASI
2019



Name : Mochamad Teguh Kurniawan, ST.,MT.
Born : Majalengka , 11 November 1986
Position : Dosen LB
Riset Intereset : Software Defined Network,
Cloud Computing, Networking
Edu : S1 Teknik Telekomunikasi IT Telkom
S2 Teknik Informatika IT Telkom
S3 Ilmu Komputer Universitas Indonesia
Address : Buahbatu Bandung
E-mail : ujangtegoeh@gmail.com
Phone : +6281321973715

KONTRAK PERKULIAH (1/2)

- ✓ Kredit : 3 SKS, 2 SKS Kuliah dan 1 SKS Praktikum
- ✓ Kuliah : TI 1&2- 2018 → Selasa 10.00 WIB → B1-106
- ✓ Kuliah : SI 1- 2018 → Selasa 13.00 WIB → B1 - 105

KONTRAK PERKULIAH (1/2)

- ✓ Perkuliahan dimulai dengan Do'a dan Diakhiri dengan Do'a Kafarotul Majlis
- ✓ Terlambat 15 menit dari Dosen masuk maka mereview materi, dikecualikan ada kejadian luar biasa (banjir, kegiatan kampus lainnya, dll)
- ✓ Keterlambatan dosen 15 menit tanpa informasi berarti tidak ada perkuliahan
- ✓ Kehadiran Minimal 75% dari Dosen
- ✓ Pengumpulan tugas ditetapkan sesuai jadwal dan keterlambatan dinilai maksimal 80%
- ✓ Dosen berhak menentukan tempat duduk atau mengeluarkan mahasiswa yang membuat kegaduhan dikelas
- ✓ Segala bentuk kecurangan contoh: tugas plagiat atau pengutipan tanpa aturan penulisan, titip absen, nyontek, diminta mengundurkan semester ini
- ✓ Makan minum boleh selama tidak mengganggu perkuliahan
- ✓ HP cukup silent
- ✓ Untuk aturan berikutnya menyesuaikan dan ditentukan kemudian

KONTRAK PERKULIAHAN (2/2)

Grading (Fix)

Nilai Grade

- | | |
|--------------------------|------|
| • Absensi | 5 % |
| • UTS | 20 % |
| • UAS | 20 % |
| • Tugas Kecil dan Quiz | 20 % |
| • Presentasi / Praktikum | 35 % |

Ada Ketua Kelas?

Download materi dan upload tugas:
<http://elen.nurulfikri.ac.id>

Pertemuan	Materi
Ke-1	Linux Kernel dan Modul
Ke-2	Proses Booting dan manajemen service
Ke-3	Manajemen Backup
Ke-4	Virtualisasi
Ke-5	Penjadwalan Tugas
Ke-6	Manajemen Logistik
Ke-7	Review All
UTS	
Ke-8	Manajemen File System 1
Ke-9	Manajemen File System 2
Ke-10	LDAP
Ke-11	Shell Programming
Ke-12	Tugas Presentasi 1
Ke-13	Tugas Presentasi 2
Ke-14	Review

S I L A B U S

QUESTION ?

NIATKAN KULIAHMU UNTUK IBADAH KEPADA ALLAH SWT

51:56

to top

وَمَا خَلَقْتُ الْجِنَّ وَالْإِنْسَ إِلَّا لِيَعْبُدُونِ ﴿٥٦﴾

Indonesian

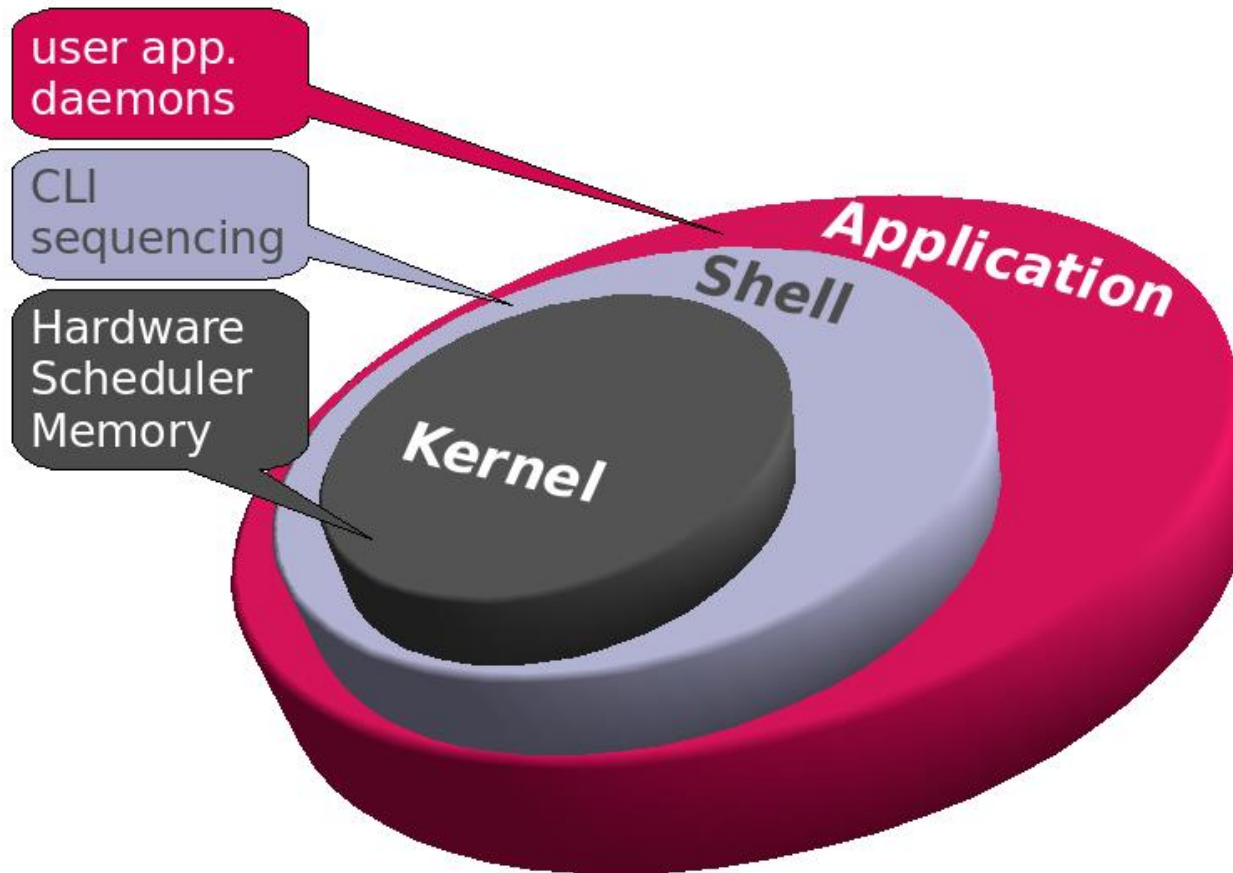
Dan aku tidak menciptakan jin dan manusia melainkan supaya mereka mengabdikan kepada-Ku.

TERIMA KASIH



Thank you very much for your kind attention

LINUX KERNEL & MODUL



APA ITU KERNEL ?



- ◆ Dalam komputasi, kernel adalah sebuah program komputer yang mengelola dan mengatur permintaan Input dan Output dari perangkat lunak (software) dan menerjemahkannya kedalam instruksi instruksi pemrosesan data untuk CPU dan komponen komponen elektronik sebuah komputer.
- ◆ Kernel adalah sebuah bagian yang mendasar (bagian utama) dari sebuah sistem operasi yang pertama kali dimuat kedalam memori saat boot dan tetap dalam memori selama sistem berjalan
- ◆ Kernel bertanggung jawab untuk melakukan manajemen memori, manajemen proses dan tugas, dan manajemen disk

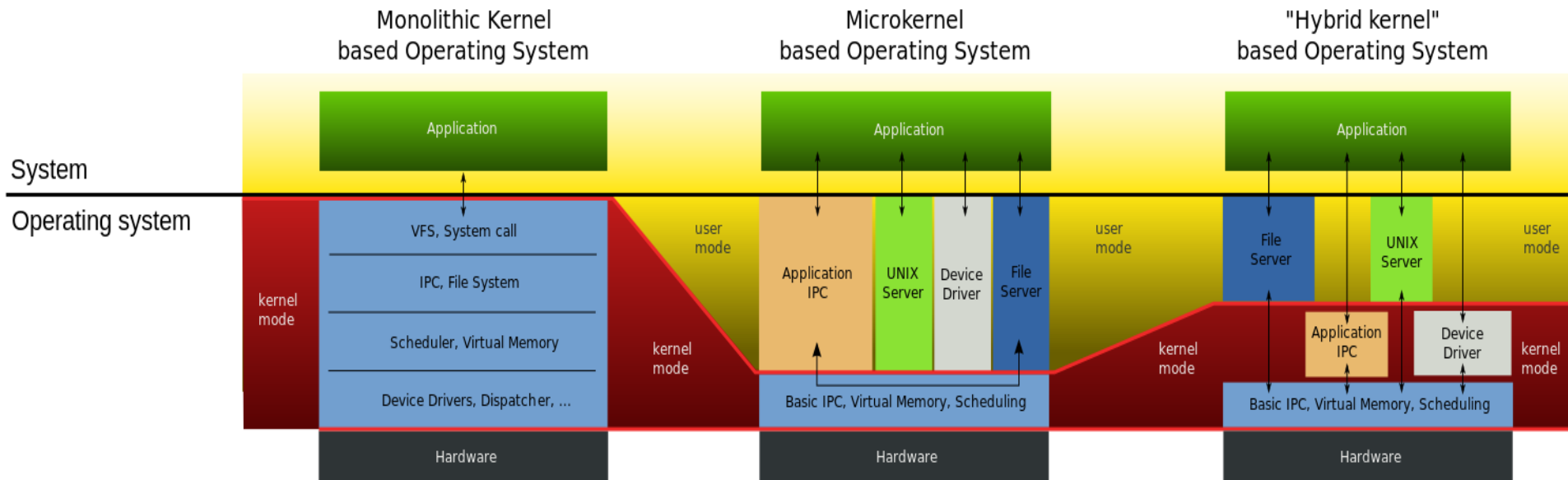
KERNEL DAN SISTEM OPERASI ?

- ★ Kernel adalah bagian dari sistem operasi yang menjembatani akses ke sumber daya sistem. Kernel bertanggung jawab untuk memungkinkan beberapa perangkat lunak untuk secara efektif berbagi perangkat keras dengan mengendalikan akses ke CPU, memori, disk I/O, dan jaringan.
- ★ Sistem operasi adalah kernel yang dilengkapi dengan perangkat lunak (aplikasi) yang memungkinkan pengguna untuk dapat melakukan sesuatu (seperti shell, compiler, editor teks, window manager, dll)

TUJUAN DESAIN KERNEL

- Performance: efficiency, speed.
 - Utilize resources to capacity with low overhead.
- Stability: robustness, resilience.
 - Uptime, graceful degradation.
- Capability: features, flexibility, compatibility.
- Security, protection.
 - Protect users from each other & system from bad users.
- Portability.
- Extensibility.

MONOLITHIC, MICRO, HYBRID KERNEL



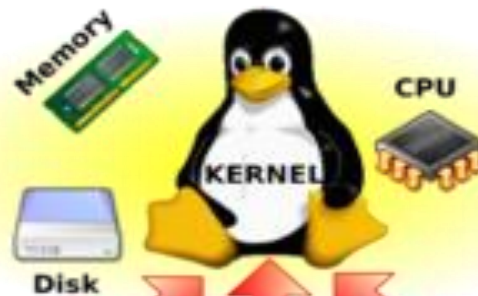
http://en.wikipedia.org/wiki/Monolithic_kernel#mediaviewer/File:OS-structure2.svg

ARCHITECTURAL APPROACHES

- Monolithic.
- Layered.
- Modularized.
- Micro-kernel.
- Virtual machine.



DEVICES



APPLICATIONS



SEJARAH KERNEL LINUX

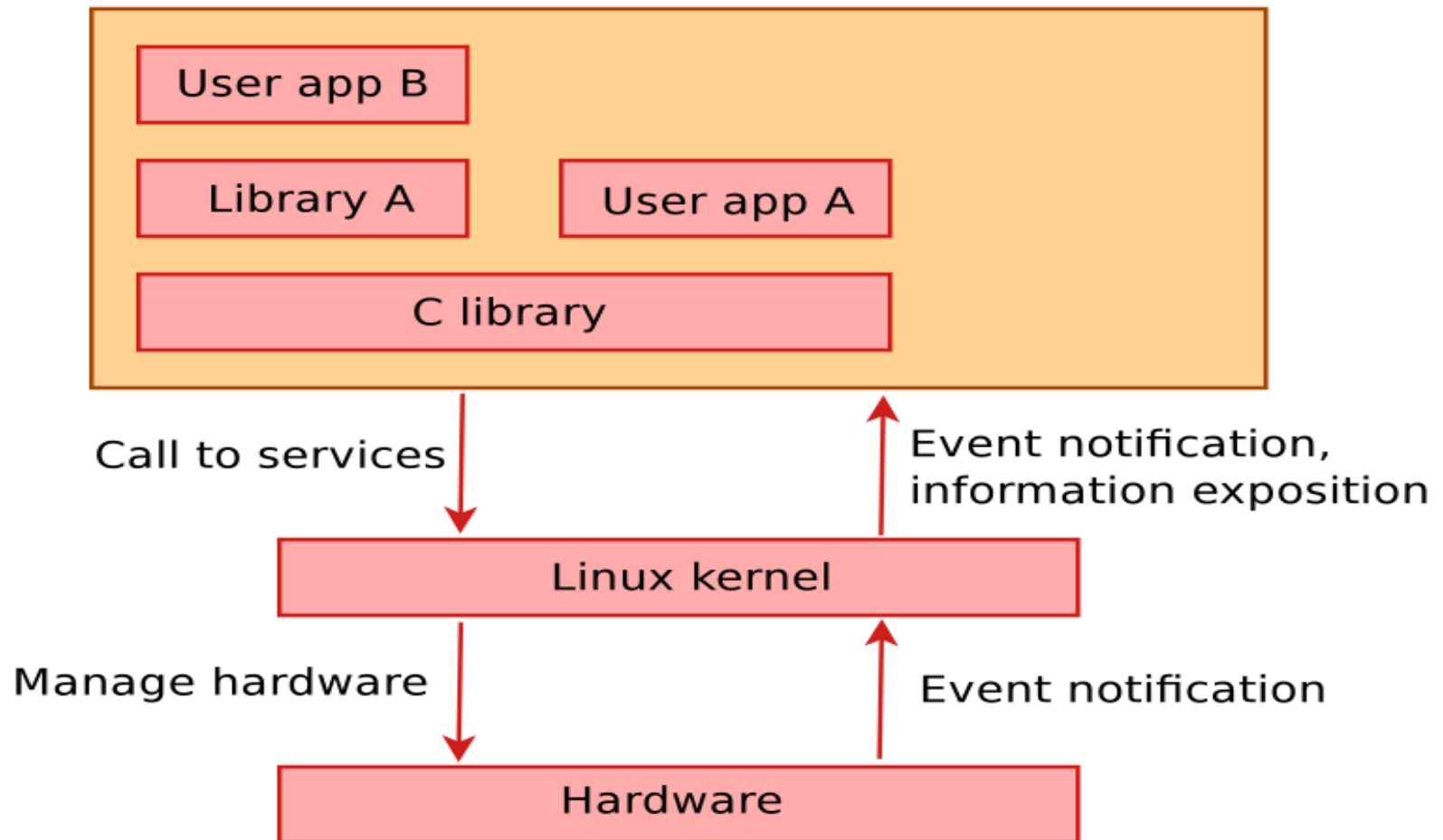
- Kernel linux diciptakan sebagai proyek hobi pada tahun 1991 oleh seorang mahasiswa Finlandia, bernama Linus Torvalds. Linux dengan cepat digunakan sebagai kernel untuk sistem operasi perangkat lunak bebas
- Linus Torvalds telah mampu membuat perkembangan para pengembang dan komunitas linux menjadi lebih besar dan dinamis. Saat ini, ratusan orang berkontribusi terhadap setiap rilis kernel linux, individu atau perusahaan besar maupun kecil.

HISTORY

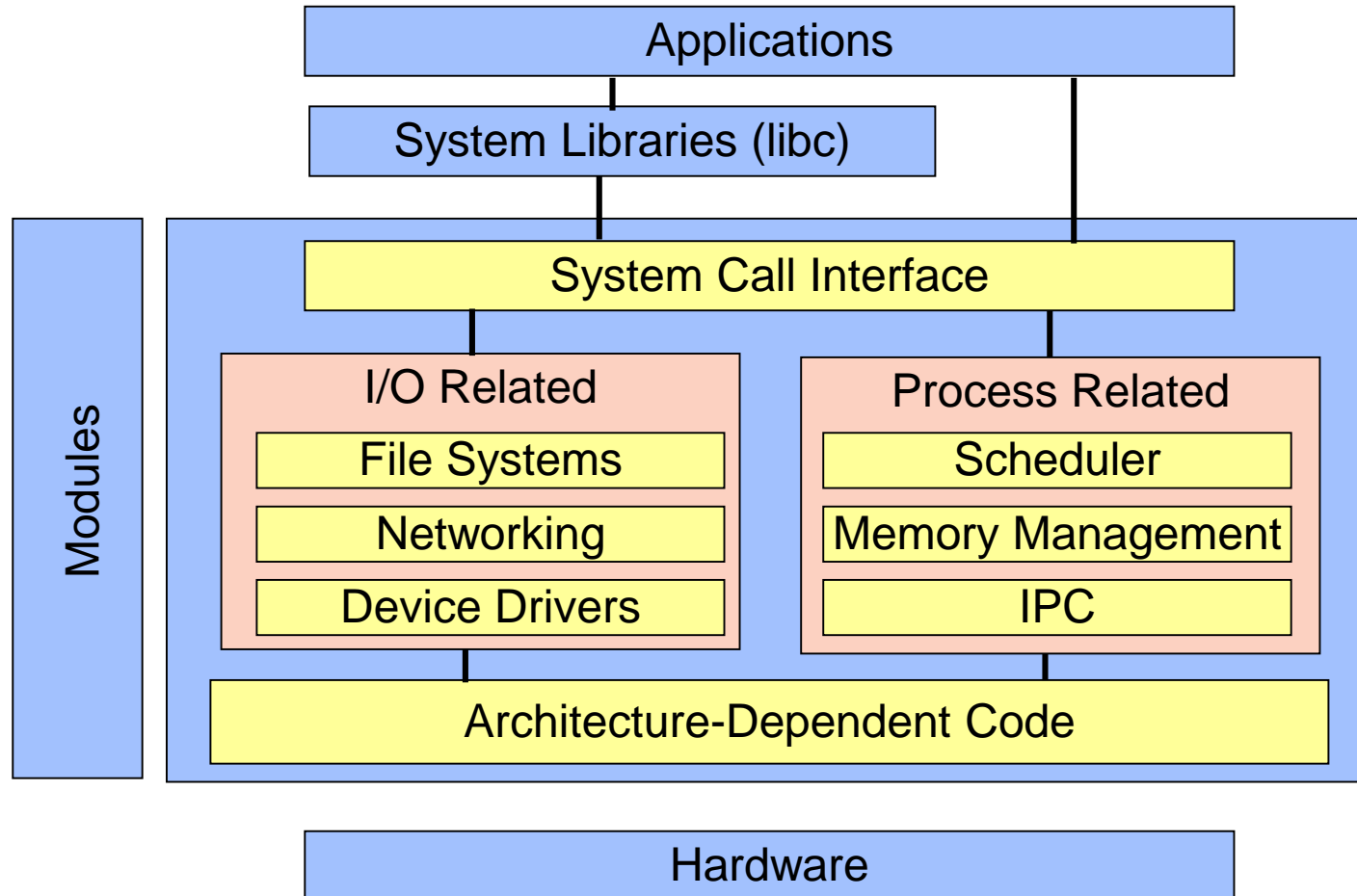


- UNIX: 1969 Thompson & Ritchie AT&T Bell Labs.
- BSD: 1978 Berkeley Software Distribution.
- Commercial Vendors: Sun, HP, IBM, SGI, DEC.
- GNU: 1984 Richard Stallman, FSF.
- POSIX: 1986 IEEE Portable Operating System unIX.
- Minix: 1987 Andy Tannenbaum.
- SVR4: 1989 AT&T and Sun.
- Linux: 1991 Linus Torvalds Intel 386 (i386).
- Open Source: GPL.

KERNEL LINUX DALAM SISTEM



EXAMPLE “CORE” KERNEL

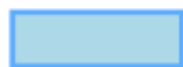
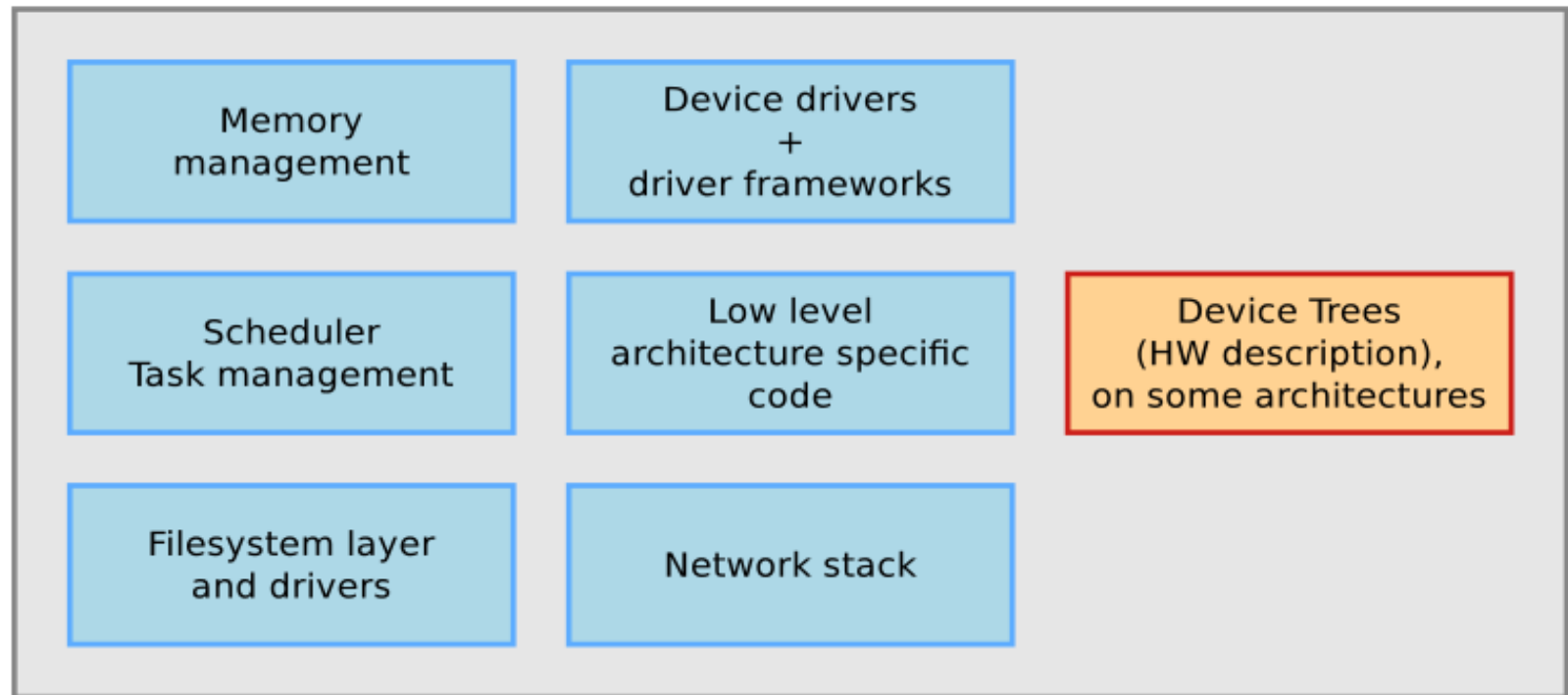


PERAN UTAMA KERNEL LINUX

- ◆ Mengelola seluruh sumber daya perangkat keras: cpu, memory, I/O
- ◆ Menyediakan sebuah set portabel, arsitektur dan API yang independen terhadap hardware untuk memungkinkan aplikasi user-space dan pustaka untuk menggunakan sumber daya perangkat keras.
- ◆ Menangani akses bersamaan dan penggunaan sumber daya perangkat keras dari aplikasi yang berbeda.
 - ◆ Contoh: sebuah antarmuka jaringan yang digunakan oleh beberapa aplikasi user-space melalui berbagai koneksi jaringan. Kernel bertanggung jawab untuk melakukan "multiplex" sumber daya perangkat keras.

DIDALAM KERNEL LINUX

Linux Kernel

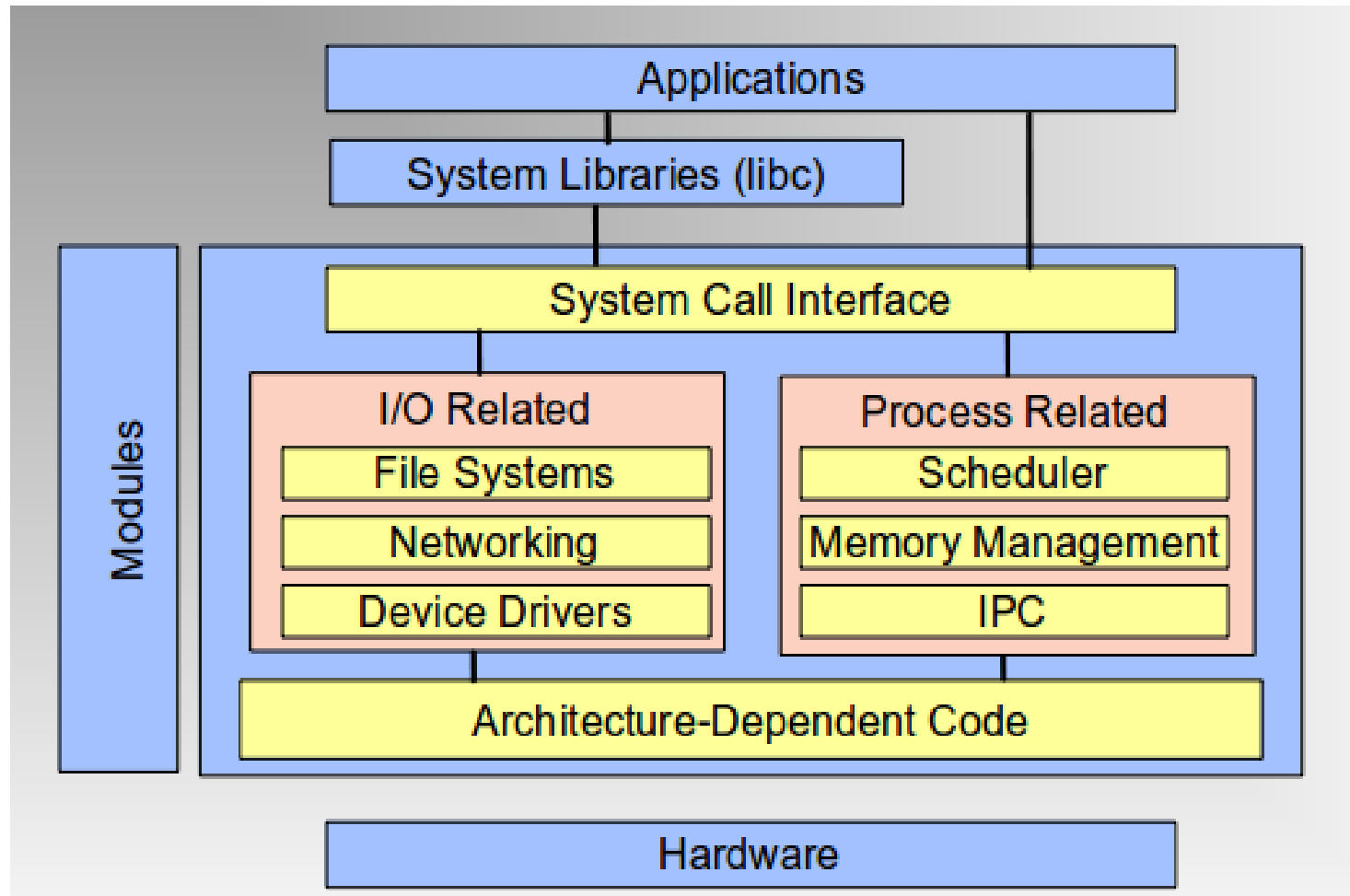


Implemented mainly in C,
a little bit of assembly.



Written in a Device Tree
specific language.

ARSITEKTUR KERNEL LINUX



LISENSI LINUX

Seluruh kode sumber kernel Linux adalah Free Software yang dirilis dibawah GNU General Public License version 2 (GPL v2).

- Lisensi ini pada dasarnya berarti bahwa:
 - Bila Anda menerima atau membeli perangkat dengan Linux di atasnya, Anda seharusnya menerima kode sumber Linux, dengan hak untuk mempelajari, memodifikasi dan mendistribusikannya.
 - Bila Anda memproduksi perangkat berbasis Linux, Anda harus melepaskan kode sumber kepada penerima, dengan hak yang sama, tanpa pembatasan

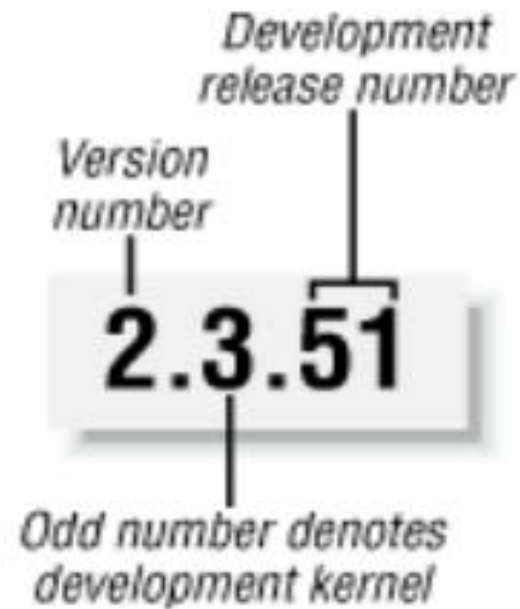
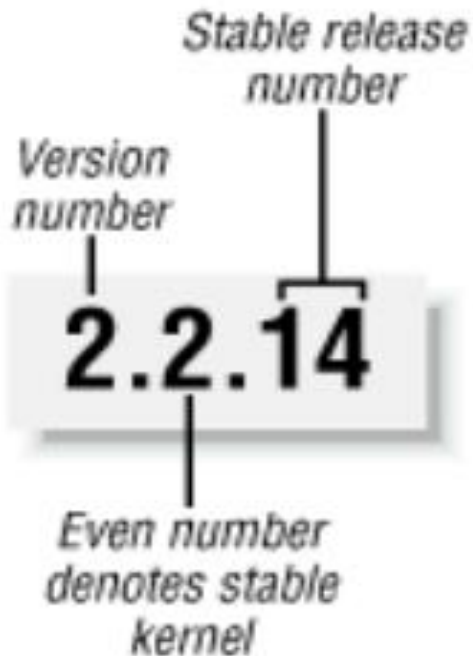
FITUR- FITUR KUNCI KERNEL LINUX

- Portabilitas dan dukungan hardware. Berjalan pada kebanyakan arsitektur
- Skalabilitas. Dapat dijalankan pada komputer super serta pada perangkat kecil (4 MB RAM sudah cukup)
- Kepatuhan terhadap standar dan interoperabilitas.
- Dukungan jaringan lengkap
- Keamanan. Ini artinya tidak bisa menyembunyikan kesalahan yang ada. Kode ditinjau oleh banyak ahli.
- Stabilitas dan keandalan
- Modularitas. Bisa hanya mencakup apa yang sistem perlukan bahkan pada saat dijalankan.
- Mudah diprogram. Anda dapat belajar dari kode yang ada. Banyak sumber daya yang berguna di internet.

ARSITEKTUR HARDWARE YANG DIDUKUNG

- Lihat direktori arch/ dalam kode sumber linux
- Minimum: 32 bit processors, dengan atau tanpa MMU, dan dukungan gcc
- 32 bit architectures (subdirektori arch/)
Contoh: arm, avr32, blackfin, m68k, microblaze, mips, score, sparc, um
- 64 bit architectures:
Contoh: alpha, arm64, ia64, sparc64, tile
- 32/64 bit architectures
Contoh: powerpc, x86, sh
- Temukan detilnya dalam kode sumber linux:
arch/<arch>/Kconfig, arch/<arch>/README, atau Documentation/<arch>/

PENOMORAN VERSI KERNEL LINUX



MENGIDENTIFIKASI VERSI KERNEL LINUX

- ➔ Untuk mengidentifikasi versi kernel linux dari sistem operasi linux yang Anda gunakan dapat dilakukan dengan cara mengetikkan perintah :
uname -r
- ➔ Atau dengan perintah:
cat /proc/version
- ➔ Dapat juga dengan Melihat isi direktori /boot, temukan file dengan nama vmlinuz-x.x.x.x

SYSTEM CALLS



- ➔ Antarmuka utama antara kernel dan userspace adalah *system call*
- ➔ Sekitar 300 system call yang menyediakan layanan-layanan utama kernel
 - ➔ Operasi File dan device, operasi jaringan, komunikasi antar proses (inter-process communication), manajemen proses, manajemen memori, memory mapping, timers, threads, synchronization, dll.
- ➔ Antarmuka system call ini dibungkus oleh pustaka C, dan aplikasi userspace biasanya tidak pernah membuat system call langsung melainkan menggunakan fungsi pustaka C yang sesuai

VIRTUAL FILESYSTEMS



- Linux membuat informasi sistem dan kernel tersedia dalam userspace melalui virtual filesystem
- Virtual filesystem memungkinkan aplikasi untuk melihat direktori dan file yang tidak ada pada setiap penyimpanan nyata: mereka diciptakan *on the fly* oleh kernel
- Dua virtual filesystem yang penting adalah:
 - proc, biasanya di mount ke /proc. Berisi informasi informasi berkaitan dg sistem operasi (seperti: parameter parameter manajemen memory, proses-proses, dll)
 - sys, biasanya di mount ke /sys. Merepresentasikan sistem sebagai sekumpulan device dan bus. Mengandung informasi tentang device dan bus.

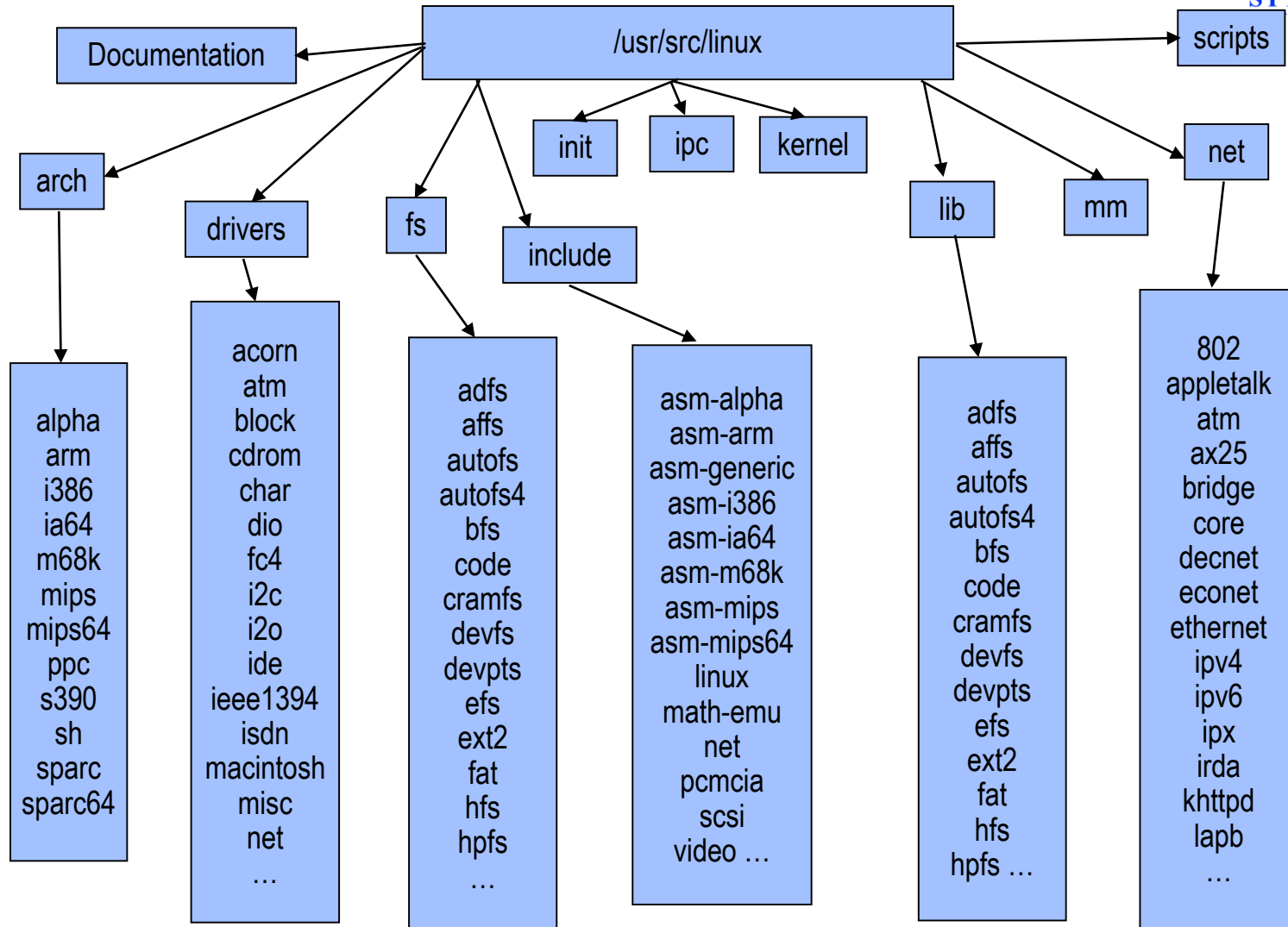
KODE SUMBER KERNEL

- Versi official kernel linux di rilis oleh Linus Torvalds tersedia di <http://www.kernel.org>
- Kode sumber kernel Linux versi 3.10:
 - Raw size: 573 MB (43,000 files, approx 15,800,000 lines)
 - gzip compressed tar archive: 105 MB
 - bzip2 compressed tar archive: 83 MB (better)
 - xz compressed tar archive: 69 MB (best)
- Mengapa ukuran kode sumber kernel linux begitu besar ?
 - Karena menyertakan banyak sekali driver driver perangkat keras (ribuan), mendukung banyak protokol jaringan, mendukung banyak arsitektur dan filesystem

LINUX SOURCE TREE LAYOUT



STT TERPADU
RUL FIKRI



MODUL KERNEL



Meskipun "monolitik," dalam arti bahwa seluruh kernel berjalan dalam satu ruang alamat, kernel Linux adalah modular, mendukung penyisipan (insertion) dan penghapusan (removal) kode secara dinamis dari dirinya sendiri pada saat runtime. Subrutin terkait, data, entry dan exit point dikelompokkan bersama ke dalam sebuah image biner tunggal, sebuah obyek kernel loadable, yang disebut modul.

MODUL KERNEL



Dukungan untuk modul-modul memungkinkan sistem untuk hanya memiliki image kernel dasar yang minimal, dengan fitur-fitur opsional dan driver-driver yang disertakan melalui loadable, sebagai obyek yang terpisah.

Modul-modul juga memungkinkan penghapusan dan pemuatan ulang kode kernel, memfasilitasi debugging, dan memungkinkan untuk permintaan pemuatan driver-driver baru dalam menanggapi hot plugging perangkat baru

MANFAAT MODUL



- ★ Module mempermudah pengembangan driver-driver tanpa reboot: load, test, unload, rebuild, load...
- ★ Berguna untuk menjaga agar ukuran image kernel minimum.
- ★ Berguna mengurangi waktu boot: tidak perlu menghabiskan waktu untuk inialisai device dan fitur fitur kernel yang hanya kita butuhkan di waktu akan datang

DEPENDENSI MODUL



- ★ Beberapa modul kernel dapat bergantung pada modulmodul lainnya, yang butuh dimuat (load) lebih dahulu.
 - ★ Contoh: modul usb-storage bergantung pada modulmodul scsi_mod, libusual dan usbcore .
 - ★ Dependensi digambarkan dalam file `/lib/modules/<kernel-version>/modules.dep`
- File ini dihasilkan ketika kita menjalankan perintah `make modules_install` setelah membangun (build) kernel.

LOG KERNEL



- Ketika modul baru dimuat (load), informasi terkait tersedia dalam log kernel
 - Kernel menjaga pesan-pesanya dalam sebuah circular buffer (agar tidak banyak mengkonsumsi memori dengan banyaknya pesan)
 - Pesan pesan log kernel tersedia melalui perintah dmesg (diagnostic message)
 - Pesan pesan log kernel juga ditampilkan ke sistem konsol
 - Perhatikan, kita juga dapat menulis pesan ke log kernel melalui user-space:

```
# echo "Debug info" > /dev/kmsg
```

UTILITAS MODUL



- Melihat modul modul yang sedang dimuat:

lsmod

- Mendapatkan informasi tentang modul:

modinfo <module_name> atau modinfo <module_path>.ko

- Memuat modul :

insmod <module_path>.ko

- Melepas modul:

rmmod <module_name>

- Memuat dan melepas modul dengan modprobe

- *modprobe <modul_name> atau modprobe -r <modul_name>*

LINUX/ARCH

- Subdirectories for each current port.
- Each contains **kernel**, **lib**, **mm**, **boot** and other directories whose contents override code stubs in architecture independent code.
- **lib** contains highly-optimized common utility routines such as memcpy, checksums, etc.
- **arch** as of 2.4:
 - alpha, arm, i386, ia64, m68k, mips, mips64.
 - ppc, s390, sh, sparc, sparc64.

LINUX/DRIVERS

- Largest amount of code in the kernel tree (~1.5M).
- device, bus, platform and general directories.
- drivers/char – n_tty.c is the default line discipline.
- drivers/block – elevator.c, genhd.c, linear.c, ll_rw_blk.c, raidN.c.
- drivers/net – specific drivers and general routines Space.c and net_init.c.
- drivers/scsi – scsi_*.c files are generic; sd.c (disk), sr.c (CD-ROM), st.c (tape), sg.c (generic).
- General:
 - cdrom, ide, isdn, parport, pcmcia, pnp, sound, telephony, video.
- Buses – fc4, i2c, nubus, pci, sbus, tc, usb.
- Platforms – acorn, macintosh, s390, sgi.