Digital Systems L2 – Computer Systems

Fang Yuan

School of Electronics Science and Engineering
Nanjing University
Nanjing 210046

2022

Outline

Embedded Systems

- Conventional computer, based on the structure, performance and scale
- Embedded (application specific) vs. general purpose computer:
  - General purpose computer (for example, PC)
  - Embedded system is equipped in machinery, products and many other electronic systems.
  - Large amount of embedded systems can be found around our daily life.

Concepts About Embedded Systems

- Application specific computer systems
- Embedded products.

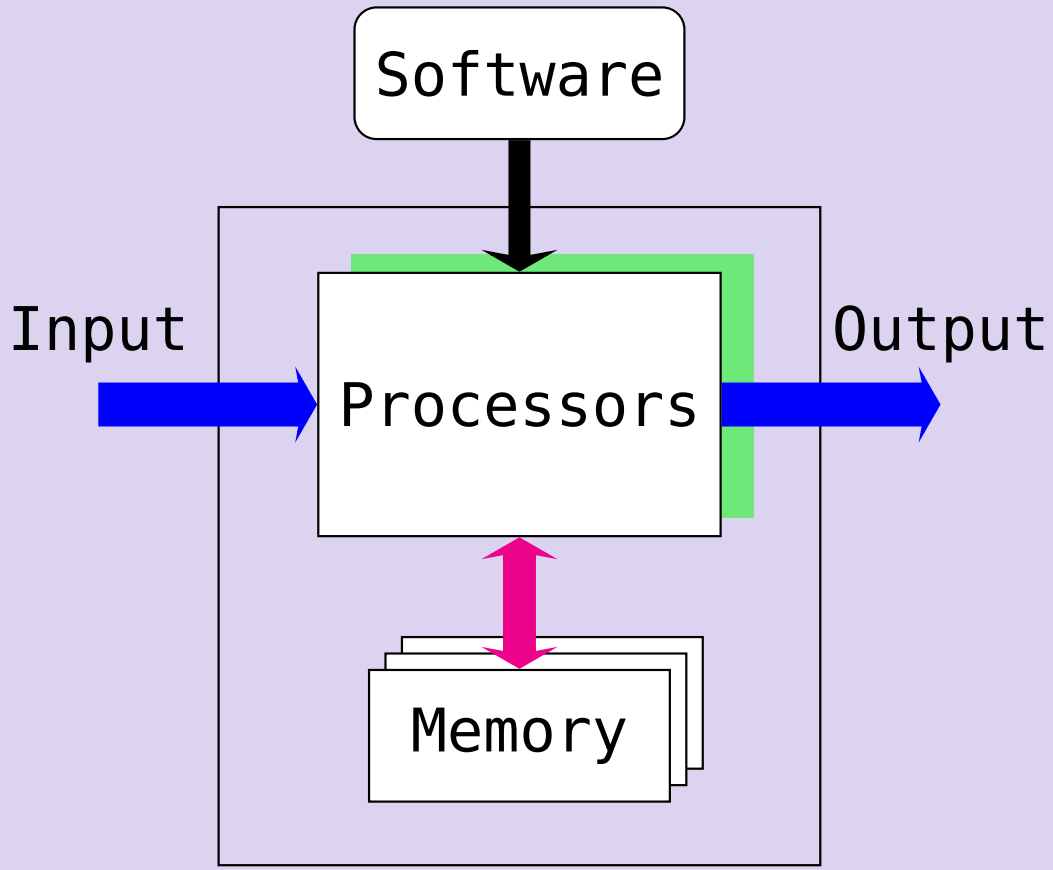
The concept *Embedded Systems* includes:

- User-oriented, products-oriented, application-oriented.
- Combination of computer techniques, electronic techniques, semi-conduct techniques and application fields.
- Well designed hardware and software system to fit its performance, costs, sizes, etc.
- Powered by embedded operating systems.

Components of Embedded Systems

Processor, memory, I/O devices and software.

Structue of an Embedded Computer



Features of Embedded System

Embedded system

- Small kernel
  Used in small equipments with limited resources, tiny OS-kernel
- Specific
  Characteristic, well organized software and hardware. System porting
- Dedicated system: size, cost, power...
- Operating system: Real-time, Multi-task.

Developing tools and software platform are important.

Embedded processors

- Micro-Controller Unit,MCU
  Small footprint and lower cost with abundant on-chip peripherals, mainly used in industrial control system
- Digital Signal Processor,DSP
  Special designing for the *digital signal processing* application.
- Micro-Processor Unit,MPU
- System on Chip,SoC
  An integrated circuit that contains a computer engine, memory and logic on a single chip.

Software Example

例

```
for (i = 0; i < TAPS; i++)
        prod += (*x++) * (*y++)
```

General Purpose Processors

```
LD *x, reg1
LD *y, reg2
MPY reg1, reg2, reg3
ADD reg3, reg4, reg4
ADD 1, x, x
ADD 1, y, y
```

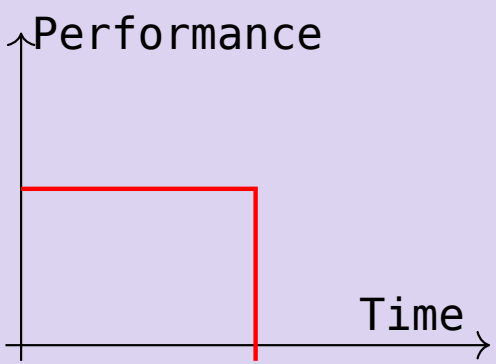
Digital Signal Processors

```
MAC (*x++), (*y++),reg4
;*** OVER ***
```

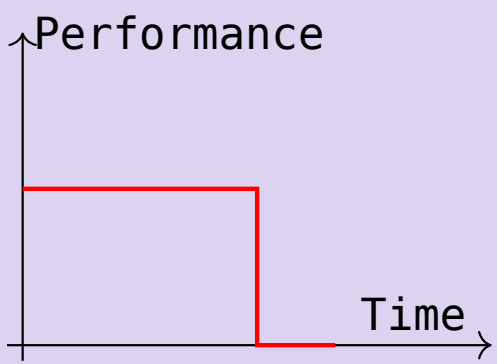
Real Time Operating System

$$\text{OS} \begin{cases} \text{Real-time} \begin{cases} \text{Hard real time} \\ \text{Soft real time} \end{cases} \\ \text{Time-sharing} \end{cases}$$

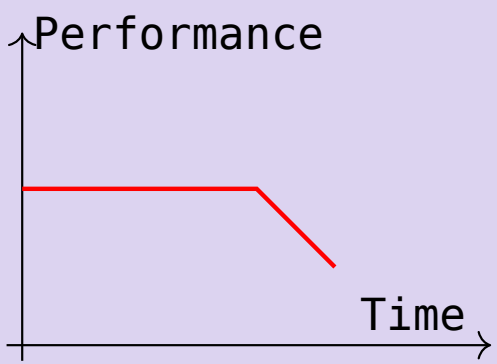
Hard real-time



Firm real-time



Soft real-time



Embedded Applications

- Industrial controls, power supply, security system, monitoring. Oil and petrol chemical, etc.
- Traffic management and control. Navigation, GPS, etc.
- Appliances such as, frig, air-conditioner and their networks
- POS, ATM, IC card,...
- Environment monitoring (Pollution, Earthquake, Weather forecasting), etc.
- Robots,...
- ...

More and more 32bit (even 64bit) processors used in embedded systems.

Developing techniques of embedded applications

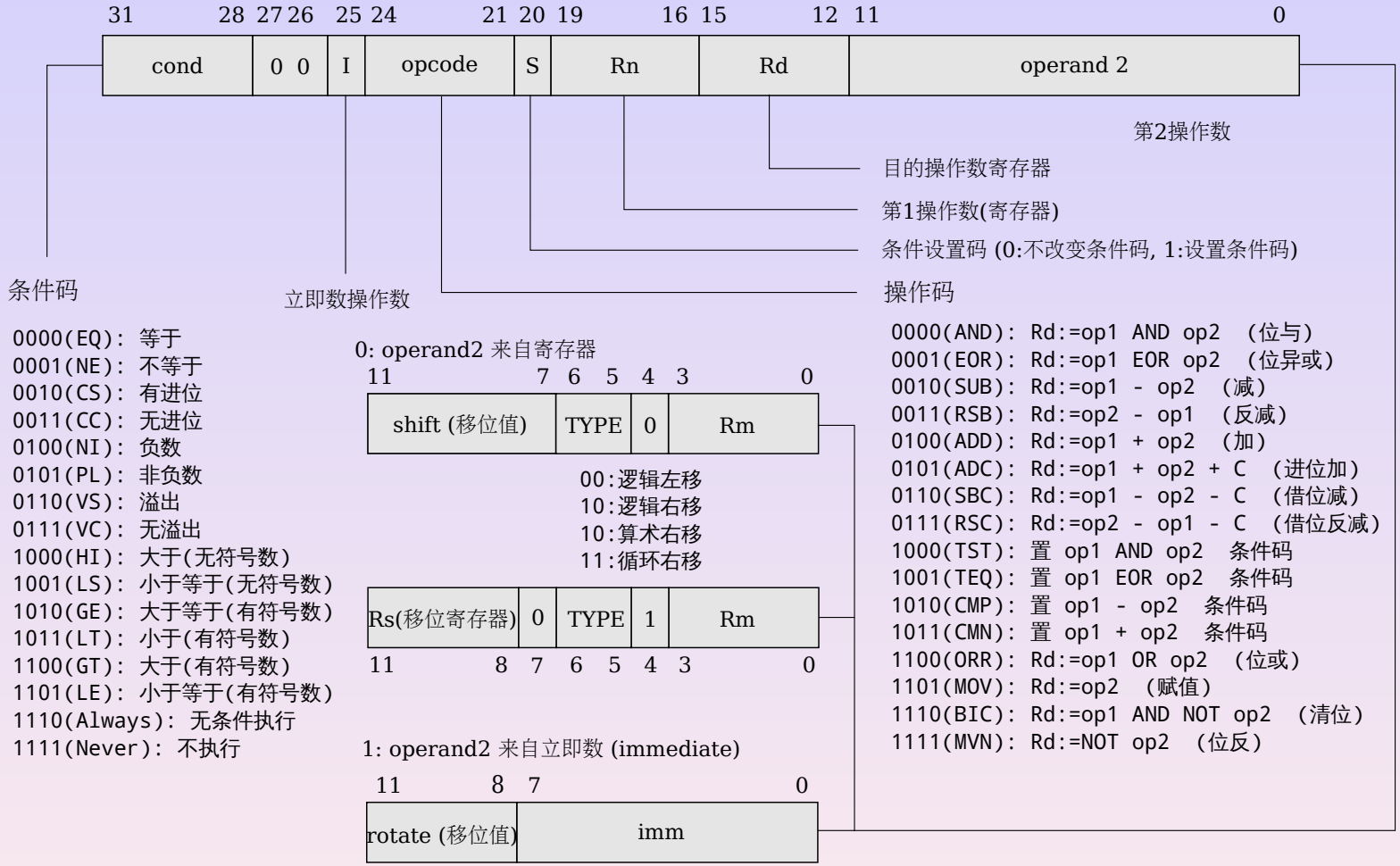
- The developing of embedded system techniques
  - Multi-functional, flexibly
  - More network supports
  - User interface(graphics, multimedia)
- Required developing skills:
  - Hardware and software platforms.
  - Computer knowledges on both hardware and software.
  - Application backgrounds.

RISC Machines in Embedded Processors

RISC vs. CISC

	CISC	RISC
Cost	higher(complexed HW)	lower
Performance	instruction-related	pipelines
Addressing modes	more	simple
instruction format	variable length	fixed-length
Control unit	micro-code	single cycle
Hardware utility	complex	fewer transistors
Registers	less	more

Arm Instruction Set



GCD with Arm Instruction Set

```
addr.    code        mnemonic
100:     e1500001    CMP     R0, R1
104:     c0400001    SUBGT   R0, R0, R1
108:     d0411000    SUBLE   R1, R1, R0
10c:     1afffffb    BNE     100
```

Commercial OSes

- VxWorks (WindRiver, 1983), Robust, hard realtime.
- Windows Embedded, easy-to-use
  - Windows CE3.0(1992), CE5.0(OpenSource, 2004)
  - Windows NT Embedded 4.0
  - Win2000 with Server Appliance Kit
- pSOS (ISI Inc., WindRiver)
- Palm OS (3COM Inc., mainly used in PDA)
- OS-9
- LynxOS
- QNX, microkernel RTOS, POSIX-compliant
- μC/OS-II

Open Source OSes

- Embedded Linux (Real-Time Linux, RT-Linux, uC-Linux)
- FreeRTOS
- Real-Time Executive for Multiprocessing Systems (RTEMS).
- eCos
- QNX, UNIX-like, part POSIX

Software Platform

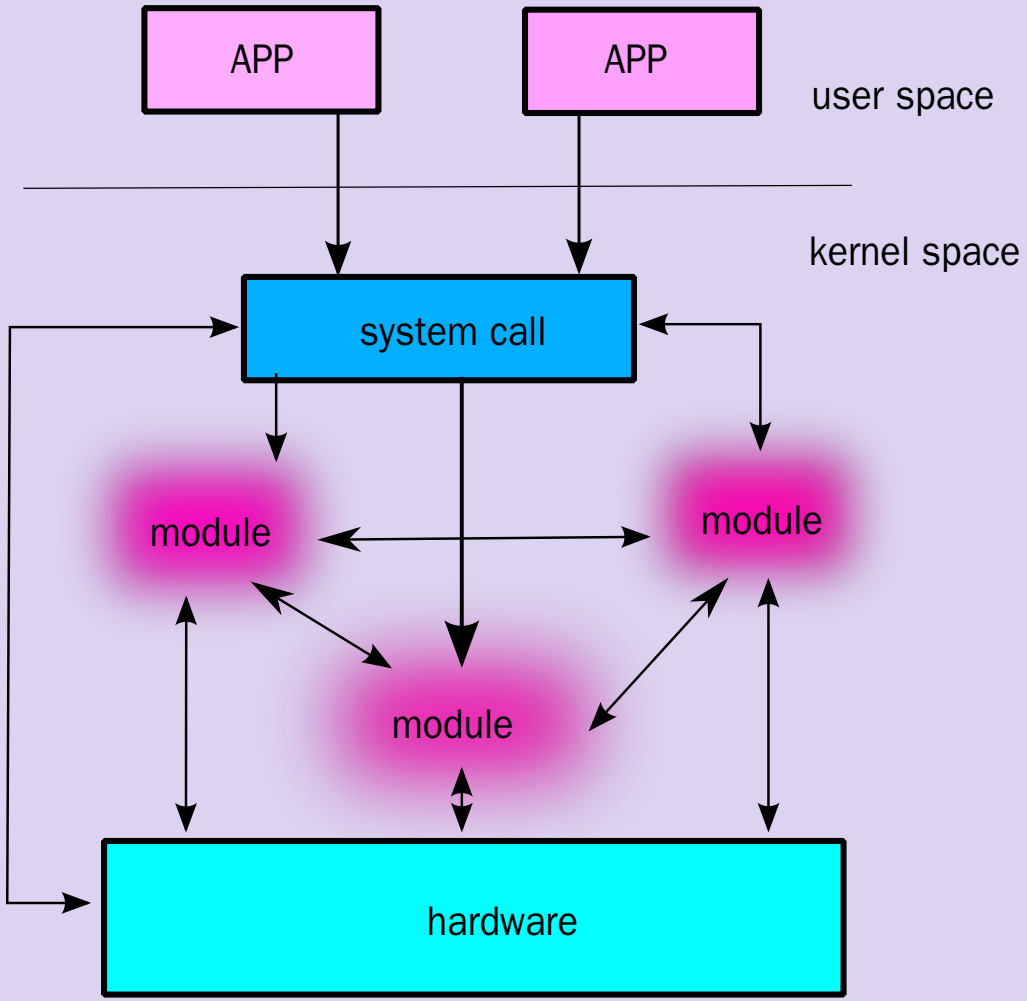
- You need to consider followings about embedded OS:
  - Network capabilities
  - Man-machine interface
  - Developing period
  - Available softwares and their capabilities.
  - Resources
  - Costs
- Supported software, including CSL (chip-support library) and BSP (board-suport package)
- Developing tools(hardware, debugger, emulator, etc.)

About OS

- Processors
- Operating systems, real-time vs. time-sharing
  - RTOS focuses on speed
  - normal OS focuses on performance and feature as a whole
- The concept *real time* of an OS deals with:
  - System response time.
  - Context-switching time.
  - Interrupt latency.
- Multi-task operating system

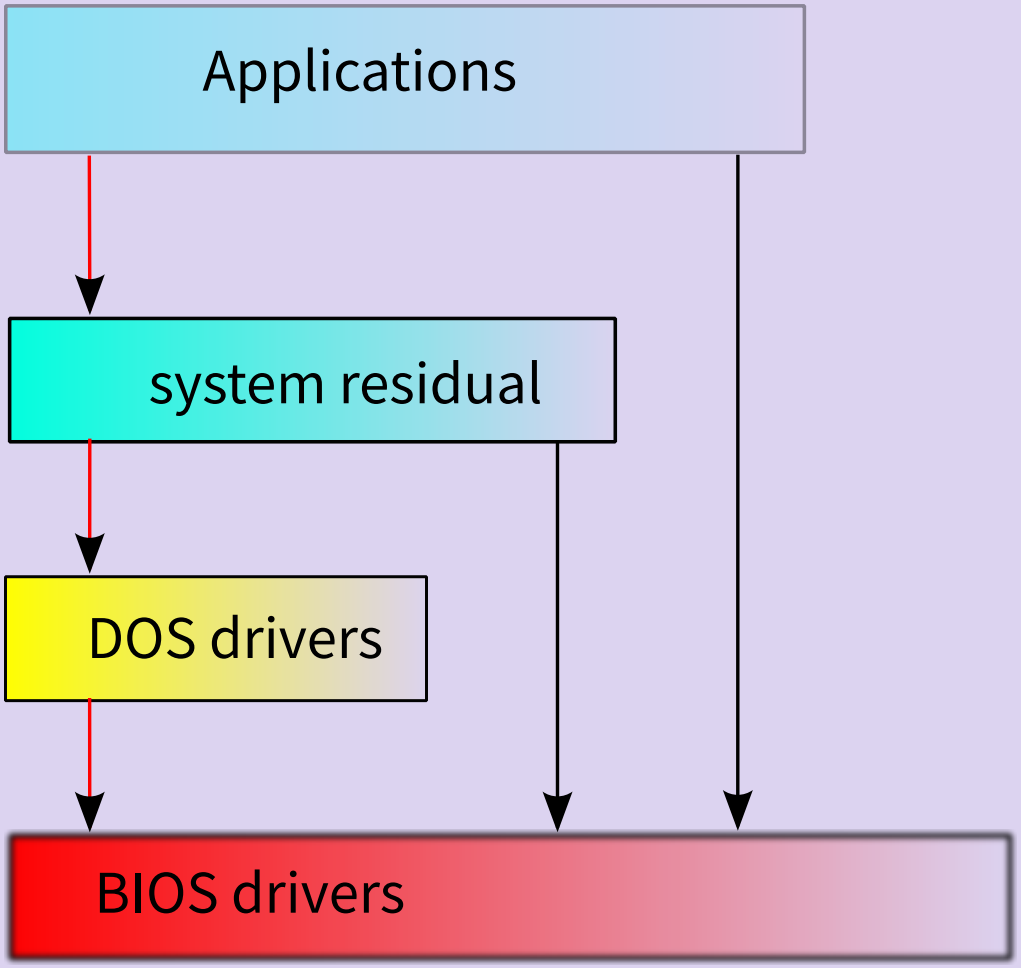
Different Layouts of OS

Moduled OS



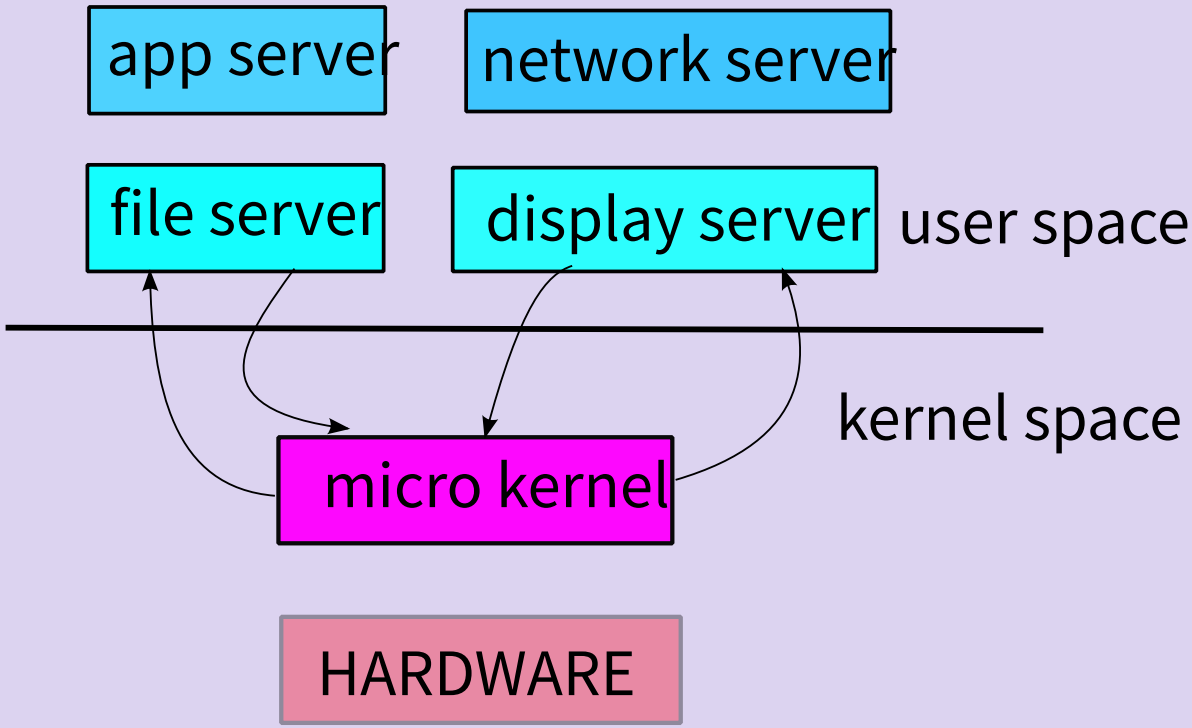
Different Layouts of OS

Layered OS



Different Layouts of OS

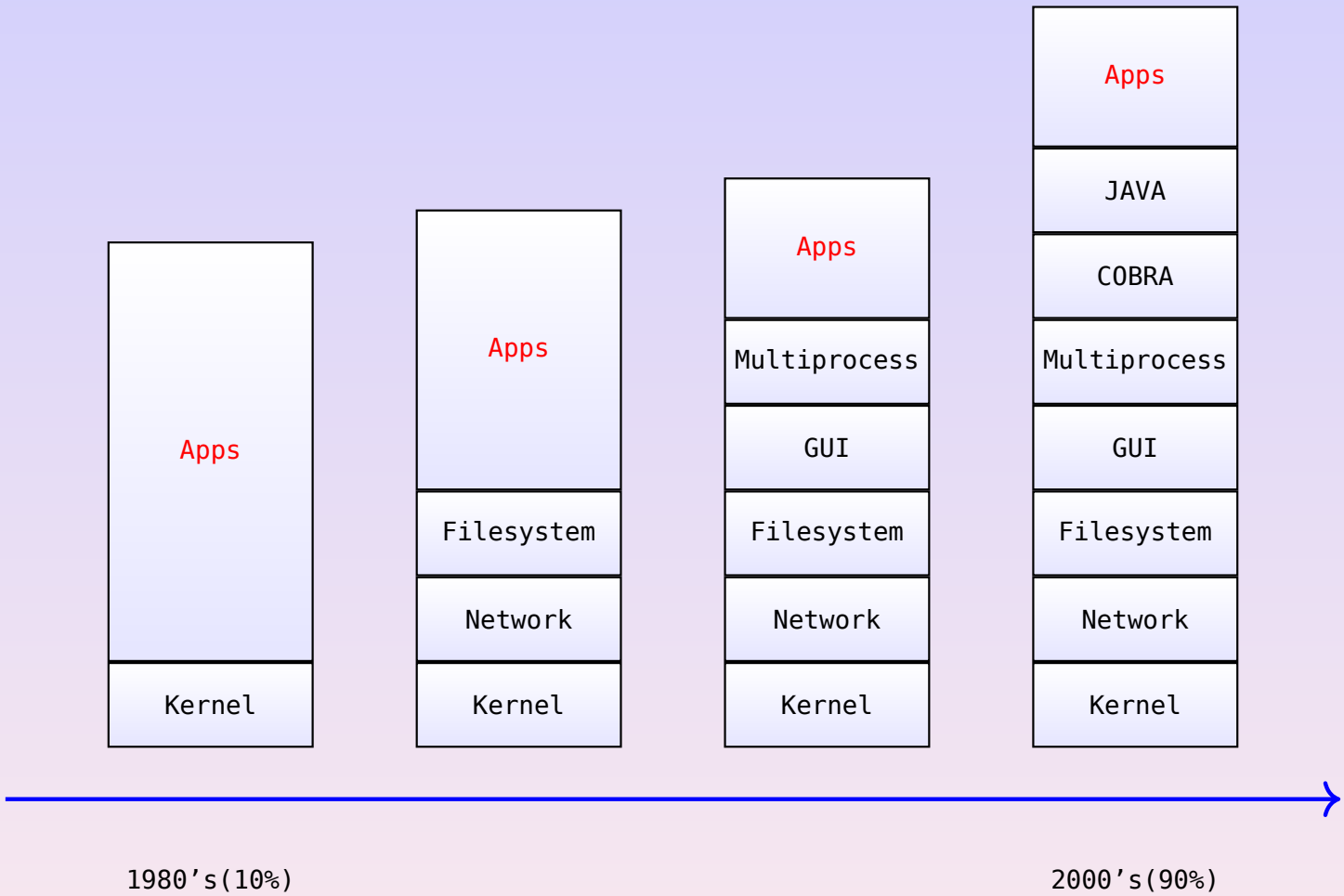
Micro kernel(C/S)



Hardware and Software Layers

Component	Resource	Software
Products		User program API (Libraries)
Embedded system	I/O devices	Drivers, OS
System board	Interfaces, buses,etc.	Board Support Package (BSP)
Processor	On-chip peripherals	Chip Support Library (CSL)

System Softwares

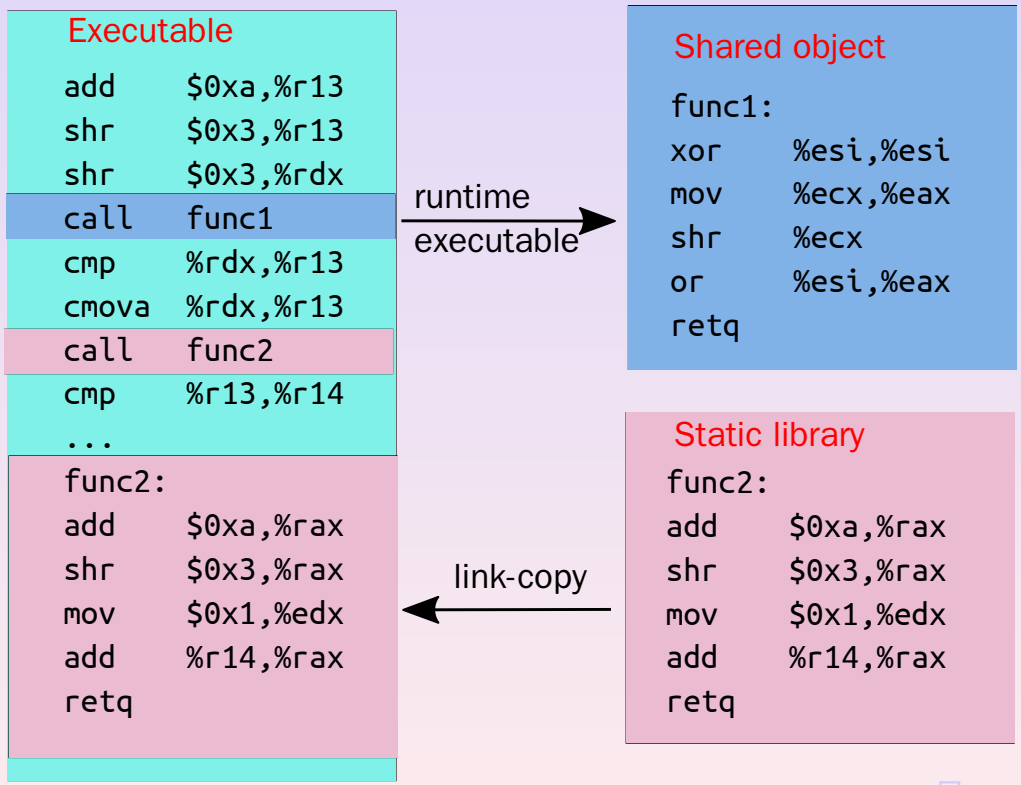


system software increases

Software Dependencies

Dependencies in Linux softwares

- Static linked libraries
- Dynamic linked libraries (shared objects)

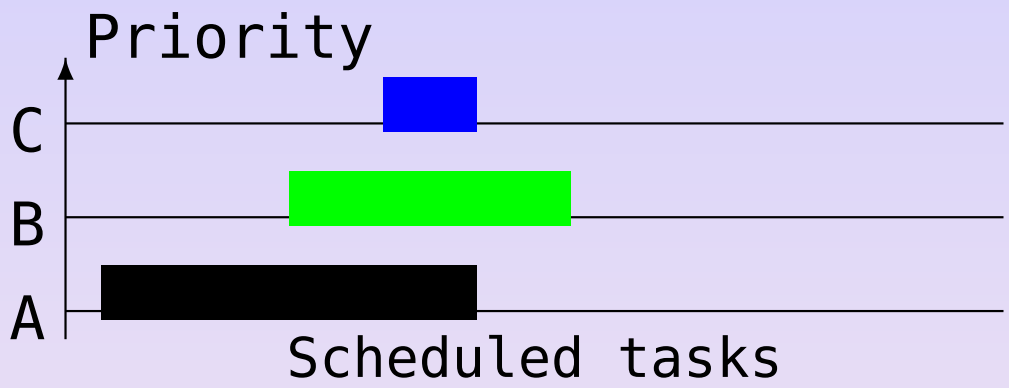


Specials on RT-OS

Real-time OS differs from time-sharing OS in the following aspects:

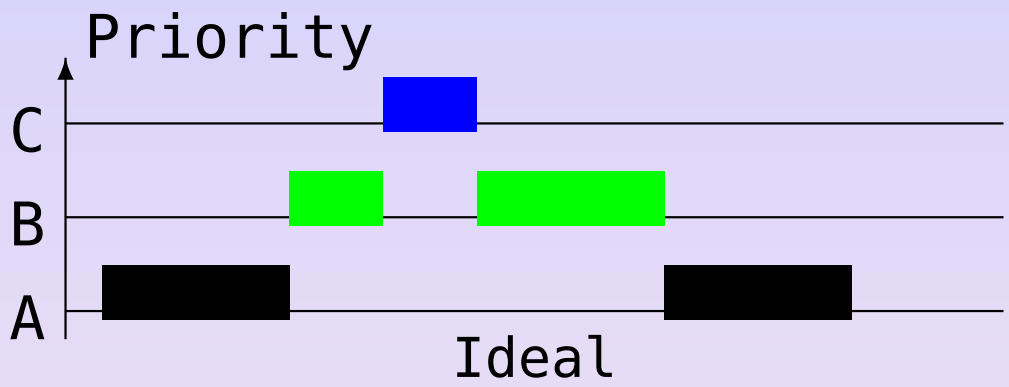
- Task scheduling
- Memory management
- Interrupts
- Resource conflicts
- Re-entrancy

Priority Inversion



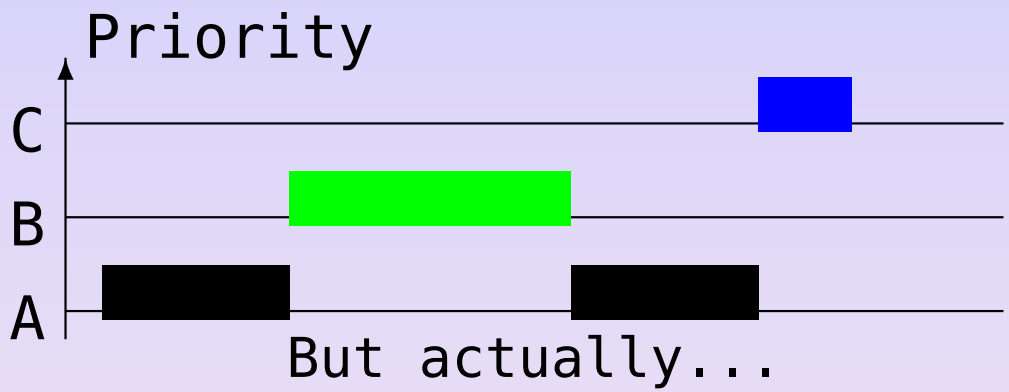
- Priority changes

Priority Inversion



- Priority changes

Priority Inversion



- Priority inheritance and ceiling promotion

Reentrancy

Consider following two pieces of code:

Global variable

```
int temp;
swap(int *a, int *b)
{
    temp= *a;
    *a = *b;
    *b = temp;
}
```

Local variable

```
swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

Left code is not reentrant.

Linux as an Embedded OS

Boot Loader

Boot loader is a piece of code which will be first executed.
In PC, boot loader is in the MBR(Master Boot Record), or in the first sector of the disk.
BIOS will load it into ram when power on.

1. WHERE is the boot loader without BIOS?
2. WHY we need boot loader, not OS itself?

Linux as an Embedded OS

Boot Loader

Boot loader is a piece of code which will be first executed.
In PC, boot loader is in the MBR(Master Boot Record), or in the first sector of the disk.
BIOS will load it into ram when power on.

1. WHERE is the boot loader without BIOS?
2. WHY we need boot loader, not OS itself?

Typical Memory Layout

A typical memory layout of the system with the bootloader, parameter structure, kernel, and filesystem might look like:

```
/* Top Of Memory */

Filesystem
Kernel
Parameter Area
Bootloader

/* Bottom Of Memory */
```

Filesystems

Filesystems that designed for IDE-like devices are not well suited for flash memory:

- Variable sector sizes.
- Poor sector-based erase/writes (limited erase lifecycle).
- Filesystem needs to be crash-proof.
- Wear levelling not supported.
- Lack of a brilliant sector management.

JFFS2(Journaling Flash File System version 2) and YAFFS (Yet Another Flash File System) are special designed for flash memory.

Tmpfs is a memory-based filesystem (reducing unnecessary flash writes to the system). It has dynamic size. Since tmpfs resides in RAM, operations to write/read/erase happen in RAM at very high speed.

Graphical User Interface

The GUI should be easy to use and pretty reliable. It needs to be lightweight, and very fast during loading.

- X11 with framebuffer support
- GTK on DirectFB
- MicroWindows
- FLTK (Fast Light ToolKit) (written in C++, not compatible with X)
- Qt/Embedded
- XFCE4 (lightweighted desktop environment)
- LXDE (Lightweight X11 Desktop Environment)
- Enligtenment (smartphone, wearables, desktop)

Summary

An embedded systems is a application-oriented computer product. The study of embedded systems inclues embedded hardware and software.

- Hardware components: processors, memory, I/O
- Software: OS, Applications
- (Firmware)

Realtime OS (or realtime tasks) required by Embedded systems.