IN.5012 **Operating Systems**
University of Fribourg, Bachelor Course, 5th Semester, Autumn Semester 2013

MINIX 3 – Introduction

Béat Hirsbrunner

MINIX 3 mascot

Main reference
Andrew S. Tanenbaum, Albert S. Woodhull
Operating Systems : Design and Implementation (The MINIX Book)
Prentice Hall, Third Edition, 2009

Lecture 1, 17 September 2013

Unix and Minix History

- **1970** : **C + Unix** (Kernigham and Ritchie, AT&T)
  - *Design of a simple programming language and operating system*
- ***1984*** *: AT&T is privatized and Unix is no longer open source !*
- ***1984 – 1994*** *: MACH (USA) and Chorus (France) OS projects*
  - *Focus: modular, distributed*
- **1987** : **MINIX** (Tanenbaum)
  - *Focus : education and open source*
- **1994** : **Linux** (Linus Torvalds)
  - *Focus : performance and open source; multi-developers*
- **1997** : **MINIX 2** (Tanenbaum)
  - *Focus : same as Minix, complete redesign and implementation*
- **2006 : MINIX 3** (Tanenbaum and Woodhull)
  - *Focus : same as Minix + embedded system (reliability, …)*

MINIX 3

- MINIX : Mini-Unix
- Unix-like operating system based on a microkernel architecture
- Open source : intend to be studied in universities
  - Very small (kernel is under 4000 lines)
  - Simple
  - Design to be readable (thousands of comments)
- Written in C,
  with some very small parts in assembly language
- Highly modular
- Reliable and secure (needed in embedded system !)

Computer Hardware Review

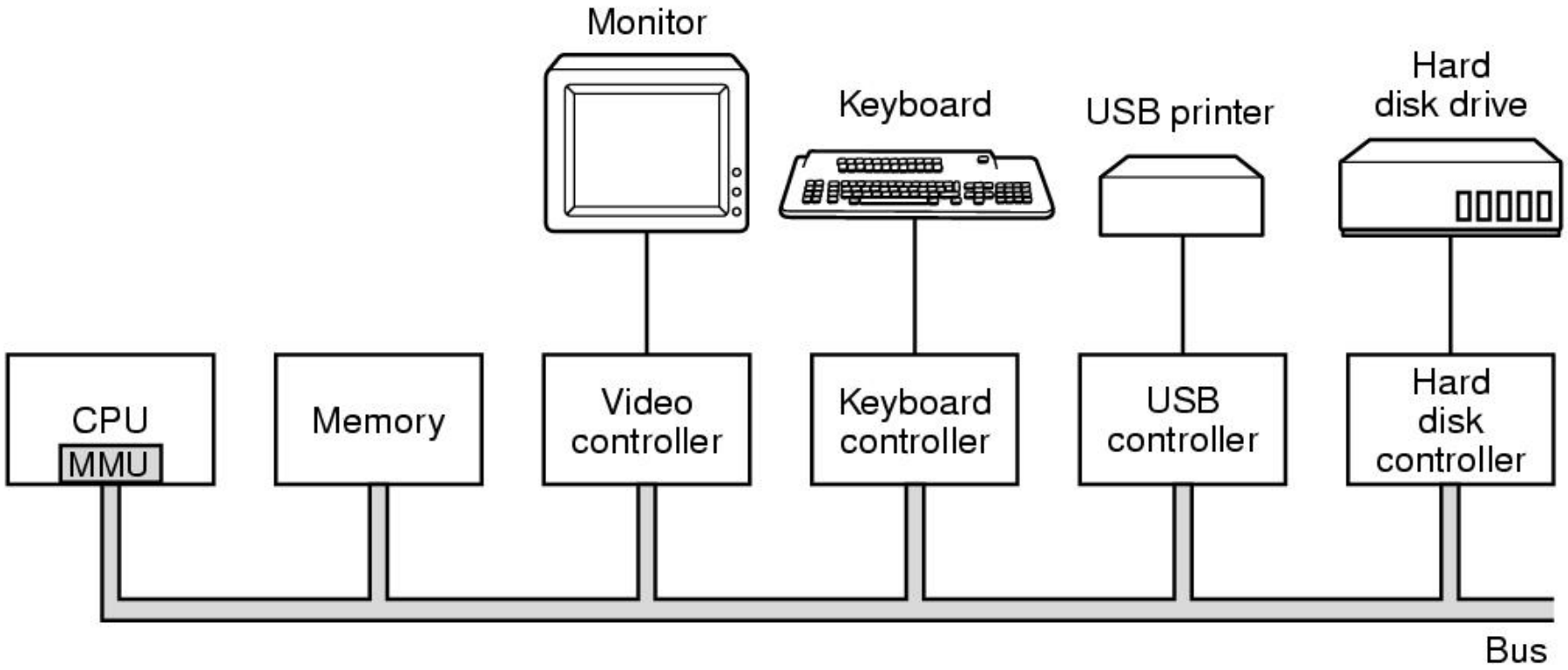

Figure 1-6. Some of the components of a simple personal computer

MINIX internal Structure

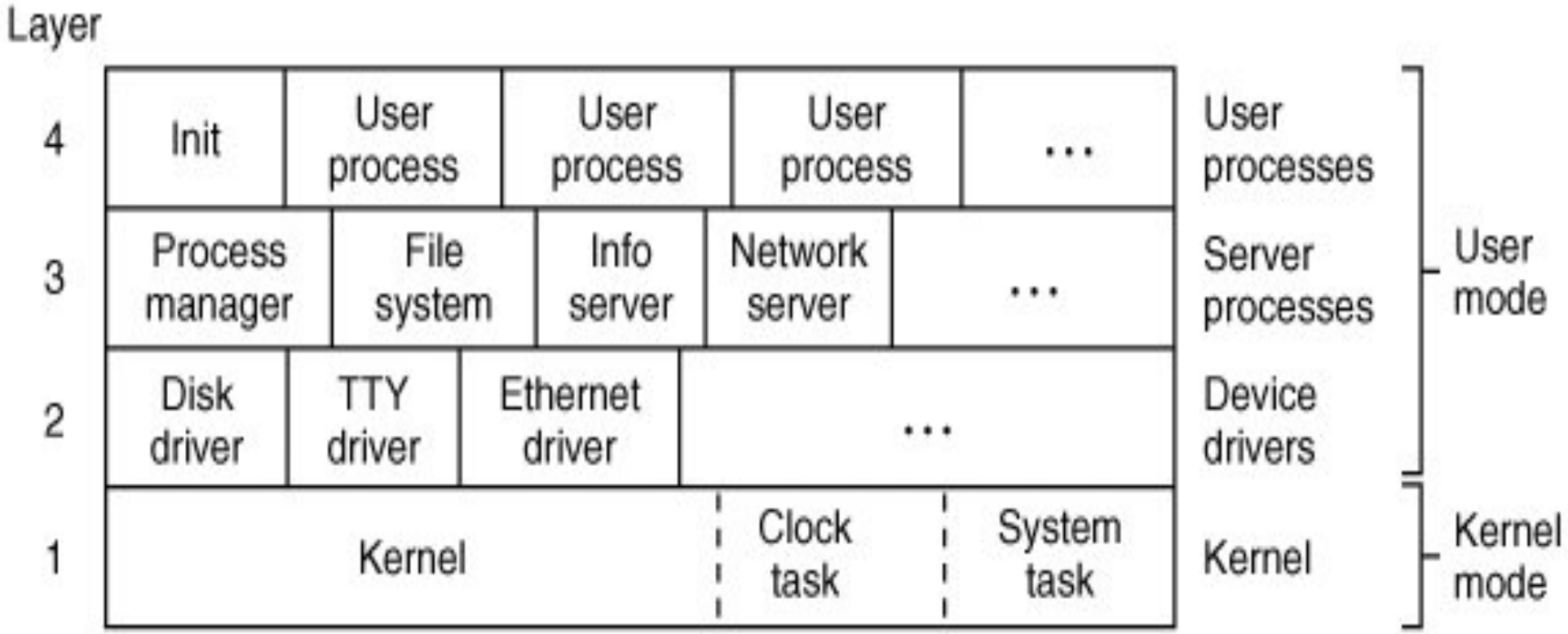

Fig 2-29. Minix 3 is structured in four layers

- Each module is executed as a process
- Communication between processes are done with the **rendez-vous message passing mechanism** (with very few exceptions: some system calls and all interrupts)

MINIX internal Structure

- **Privileges**
  - Layer 2 and 3 have privileges to make kernel calls, but not layer 4
- **Kernel module (layer 1)**
  - Schedules processes, and manages the transitions between the ready, running and blocked states
  - Handles all messages between processes
  - Access to I/O ports and interrupts
- **Layer 2: Device drivers**
  - A driver is needed for each device type, it has access to I/O ports
- **Layer 3: Server providing useful services**
  - Process manager (**PM**) and file system (**FS**) servers are essential
  - Minix 3 has a very simple main memory management: no virtual memory, no swap ! It is possible that in a future release **PM** (process/thread and memory mngt) will be separated in **PM** and **MM**.

Interprocess Communication

- Three primitives to send and receive messages
  - send(dest, &message);
  - receive(source, &message);
  - sendrec(src_dst, &message);
- Provided by a C library mechanism inside the kernel
- Rendez-vous principle
- Messages passing

MINIX startup

- **Boot program**
  - The boot image defines the processes to load in memory
  - Init is the last process to be executed in the boot sequence
- **Process tree**
  - Process Manager has pid 0
  - Init is the first user process and ancestor of all users processes
  - Kernel, system and clock tasks are not in process tree

Component	Description	Loaded by
kernel	Kernel + clock and system tasks	(in boot image)
pm	Process manager	(in boot image)
fs	File system	(in boot image)
rs	(Re)starts servers and drivers	(in boot image)
memory	RAM disk driver	(in boot image)
log	Buffers log output	(in boot image)
tty	Console and keyboard driver	(in boot image)
driver	Disk (at, bios, or floppy) driver	(in boot image)
init	parent of all user processes	(in boot image)
floppy	Floppy driver (if booted from hard disk)	/etc/rc
is	Information server (for debug dumps)	/etc/rc
cmos	Reads CMOS clock to set time	/etc/rc
random	Random number generator	/etc/rc
printer	Printer driver	/etc/rc

Organization of the Source Code

- src/kernel – layer 1 (scheduling, messages and system tasks)
- src/drivers – layer 2 (device drivers for disk, console, printer, …)
- src/servers – layer 3 (process mngr, file system, other servers)¨

- scr/lib – source code for library procedures (open, read, …)
- scr/tools – Make file and scripts for building the Minix3 system
- scr/boot – the code for booting and installing Minix 3
- …

- Each directory has its own Makefile
- Run make inside /usr/src/tools

References

- A. S. Tanenbaum, A. S. Woodhull : *« Operating Systems : Design and Implementation (The Minix Book) »*, Third Edition, Prentice Hall, 2009
- http://www.minix3.org
  - Downloads, Installation Guide, …, Research Projects
- http://en.wikipedia.org/wiki/Unix
  - Unix History
- http://en.wikipedia.org/wiki/Minix
  - Minix History
- http://en.wikipedia.org/wiki/POSIX
  - POSIX Overview
- http://en.wikipedia.org/wiki/Application_programming_interface
  - API Overview