Introduction & History
Ch1

Two Roles of an Operating System

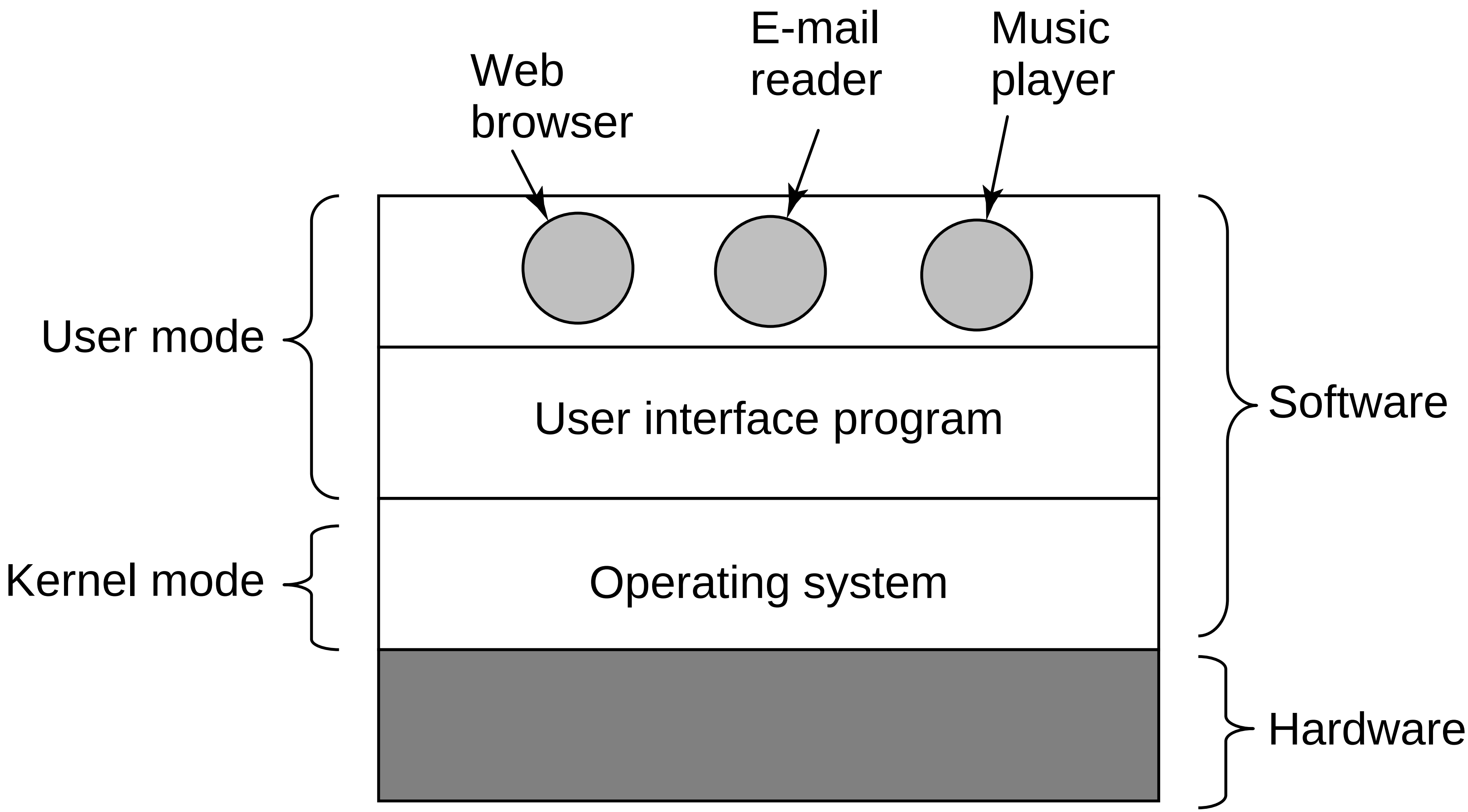
Extended Machine
or virtual machine

Device drivers, Processes, File systems, Networking protocols.

Resource Manager
Allocates and enforces

Memory, Disk space, Devices (*e.g.* printer)

Structure


Web
browser
E-mail
reader
Music
player
User mode
User interface program
Software
Kernel mode
Operating system
Hardware

First Gen 1940's and 1950's

Relays or vacuum tube hardware.
Plug board or punch card input.

Programmer runs his own program.

Reserves time on a paper sign-up sheet.

No Operating System

Problems

- Scheduling inefficiency.
- Time wasted mounting tapes and cards.

Result: Low utilization of a very expensive piece of equipment.

Second Gen: Batch Mainframes

Mid-1950's to Mid-1960's

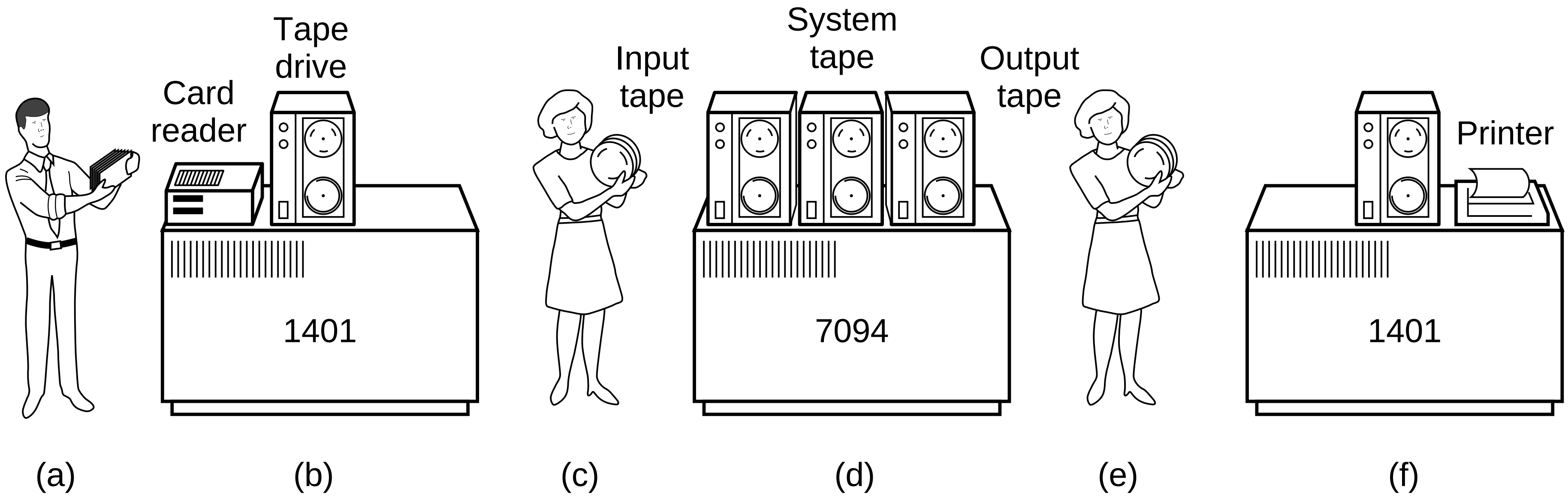
Transistorized equipment.
Punch-card input.

Users distinct from the designers and builders.

Equipment isolated and controlled by operators.

User inputs a stream of jobs, usually on cards.

Batch Systems



Third Gen: Multiprocessing

Integrated Circuits
Multiprogramming

Leader: IBM 360.

One machine for both scientific and business DP.

Multiprogramming: Leverage the mix of I/O and compute-bound jobs.
Multiple jobs in memory.
Each has its own contiguous region.

Spooling: Main machine does its own job input.

Time-sharing

Desire to return to interactive use.

Leader: Compatible Time Sharing System (CTSS)

MULTICS: A centralized "computing utility."
Advanced the field
Not a commercial success

Unix: Created by ATT scientists left w/o MULTICS
Written on spare minicomputer
Very popular in academia and elsewhere

BSD and SysV

Linux

Fourth Gen: Personal Computers

1975: Intel 8080
CP/M

1981: IBM PC
DOS

1960's: Douglas Englebert at SRI
Xerox PARC
GUI

Steve Jobs (Apple) visited PARC
Lisa
Macintosh

Fourth Gen: Personal Computers

MS influenced by Mac

Windows Windows 95 Windows 98 Windows ME

Windows NT Windows 2000 Windows XP

Vista Windows 7

Unix popular on engineering stations

Linux occasionally

X-Windows

OS Types

Mainframe   Server   Multiprocessor   PC

Handheld   Embedded

Sensor Node   Smart-Card

Real-Time
Hard   *Soft*

Hardware Components

CPU Memory
Process abstraction

I/O Devices
Drivers, clean interfaces

Storage Devices
File system abstraction

CPUs

Traditionally, there is one CPU.

Multiple-CPUs used on high-end systems.

Muliti-core systems are now the usual.

Multithreading (or hyperthreading) makes it faster for the OS to change which program is running.

CPU Registers

Program counter.

Stack pointer.

General purpose registers.

Program Status Word (PSW)
Collection of status values and flags.

Collectively the CPU state

I/O

I/O operations are slow.
CPU must start the operation then wait for completion

Polling
Loop on a status query
Are we there yet?

Interrupt-Driven
CPU start operation; device interrupts CPU on completion.

Direct Memory Access (DMA)
A DMA controller batches of I/O operations.

Interrupts

Much like an asynchronous function call.
Similar to event handlers in a GUI.

Peripheral device signals the CPU.

CPU calls the *interrupt vector* in response.

CPU saves its state, runs the vector, then restores the state.
Restoring the PC returns to calling point.

Avoids the need to poll.

Traps

CPU interrupts itself.

Program errors or exceptions.

Divide by zero

Illegal address

Undefined op code

etc

Hardware Support for the OS

Certain hardware features are required to implement a reasonable OS

Traps and Interrupts

CPU Modes

Memory Protection

Clock Interrupt

Traps and Interrupts

Interrupt vectors are part of the O/S.
Interrupts give control to the O/S.

O/S regains control when I/O completes.

<u>O/S Can let other other code run during I/O</u>

CPU Modes

User and supervisory mode.

User code uses user mode
OS uses supervisory mode

Some instructions forbidden in user mode.

Interrupts and syscall change to supervisory.

OS switches to user before return.

<u>Reserves certain operations for the OS alone.</u>

I/O Operations are Privileged

Most I/O operations are forbidden in user mode.

Gives O/S exclusive control over disk operations.

A user programs must ask the OS to perform IO for it.

Keeps user code from violating file permissions.

Memory Protection

In user mode, CPU may access only a portion of the memory.
Violations trap

Base and limit registers.
Segment codes.
Virtual memory (ch. 3).

Instructions to set or modify protection are privileged.

<u>OS sets memory limits, hardware enforces.</u>

Clock Interrupt

Periodically interrupts the CPU.

Setting the interrupt time is privileged.

Makes certain the CPU always gets control eventually.

Allows O/S to enforce CPU allocation.

System Calls

How user software talks to the OS.

Syscall instruction is a deliberate trap.

Store arguments in a standard place.
Push on the stack or store in standard registers

Execute the syscall instruction.

OS handles the operation and returns.

Operations Performed With System Calls

Create a file or directory.

Read or write a file.

Send or receive a network packet.

Create or terminate a process.

Anything that requires privileged instructions.

Structure

Monolithic

Single supervisory-mode kernel provides all O/S services.

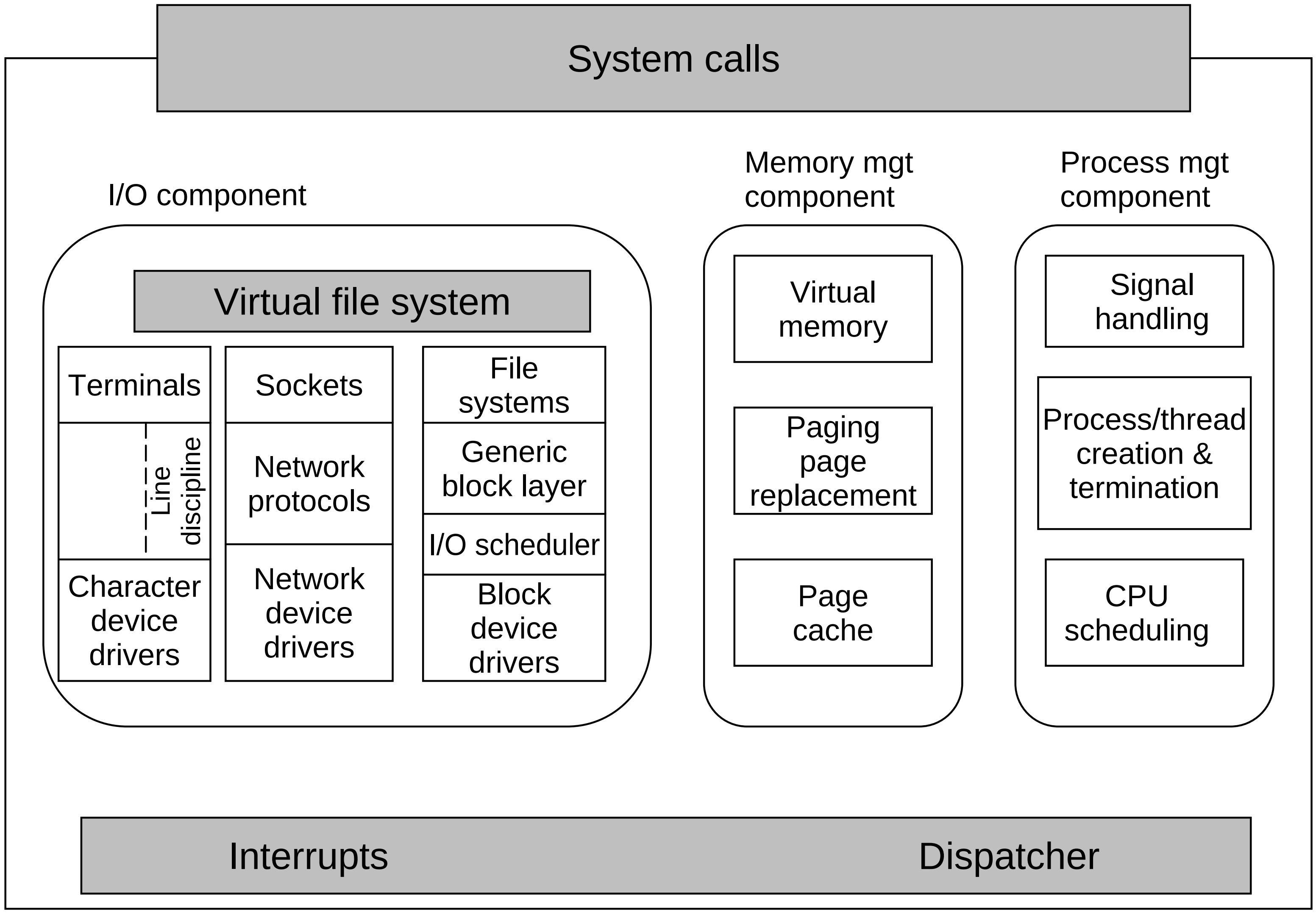
Layered

Kernel is build in layers each provided additional services to those above.

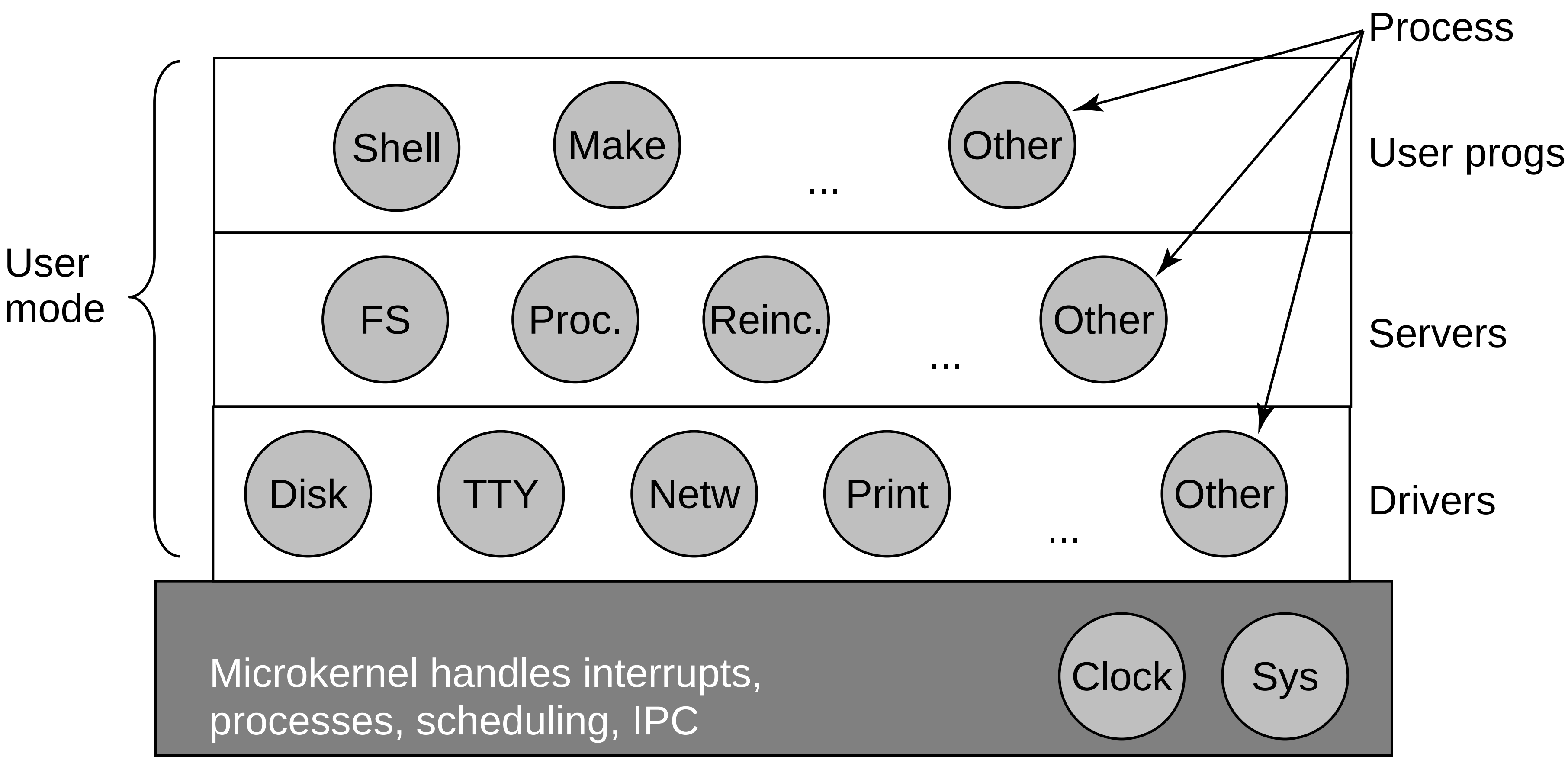
Micro-kernel

Minimal supervisory-mode kernel

Linux Layered Kernel



Micro-kernels



Micro-kernels

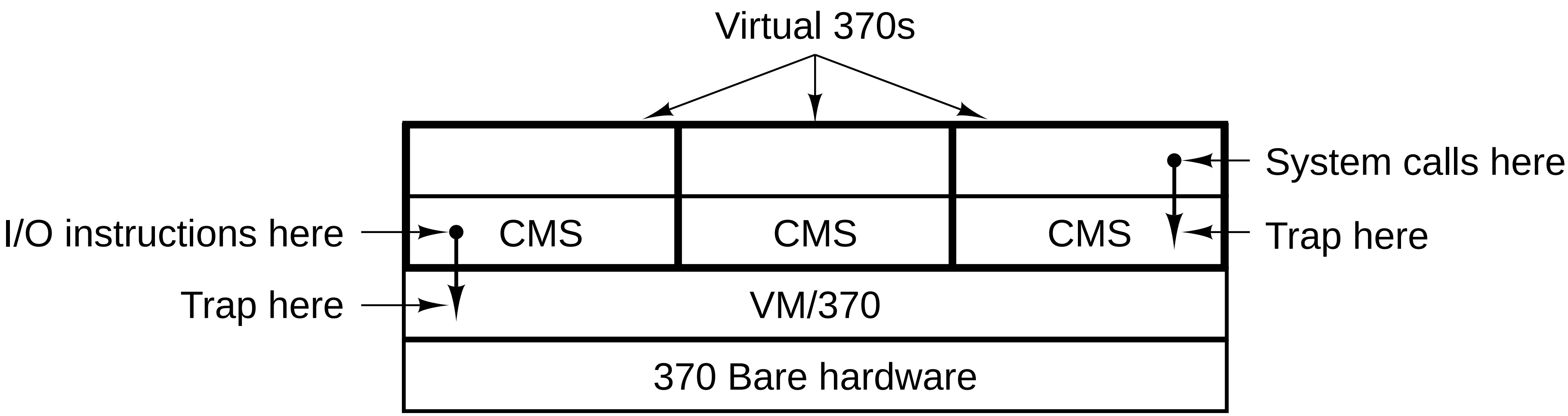
Clean, extensible design.

Message-passing easily distributed.

Has not been commercially successful for performance reasons.

Virtual Machines

VM layer divides the resources to provide exact, shrunken copies of the hardware

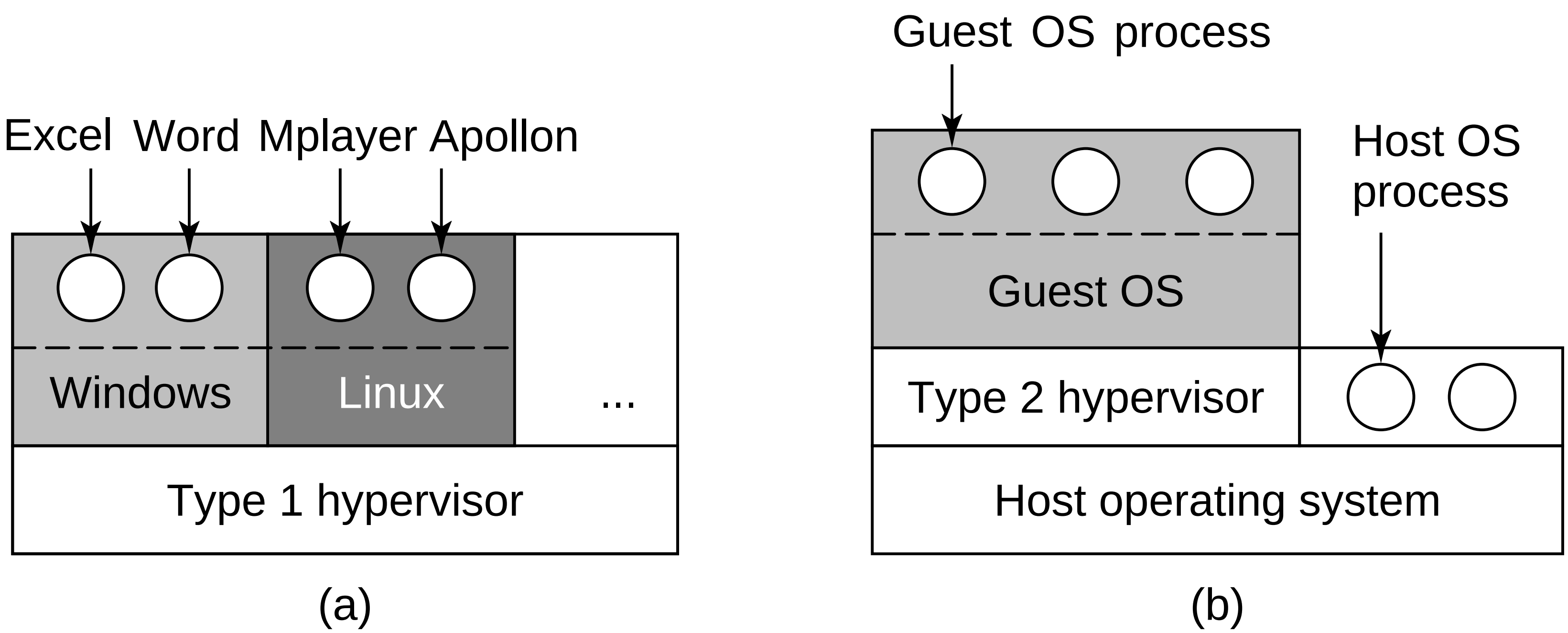


DOS mode under Windows

Two Types

Type 1: VM runs on the hardware.
Used by IBM in the 70s.

Type 2: VM runs on an OS
VMWare, Virtual Box, Virtual PC



IBM Had It Easier

VM must run in user mode and simulate privileged instructions.

Easiest of the instructions cause traps.

Pentium class machines tend to ignore privileged instructions.
Pentium VMs have to do more simulation.

Current generations translate binary by block.

Sources

Tanenbaum, *Modern Operating Systems*
(Course textbook.)