Administrative Information & General Introduction

Lectures: Prof. Neeraj Suri, Dr. Stefan Winter
Exercises: Oliver Schwahn, Habib Saissi

DEEDS (Dependable Systems & SW Group)
www.deeds.informatik.tu-darmstadt.de

Contact Information

Course homepage:
https://www.informatik.tu-darmstadt.de/deeds/teaching/wise2018/betriebssysteme

Contact Options:

- For any questions: HRZ moodle forum
- Tutor office hours (see slide on exercise administration)

Lectures

Wednesday, 15.20-16.50, S1|01 A1 (Audimax)

Number	Date	Topic	Language
1	17.10.2018	Intro	German
2	24.10.2018	Processes & IPC	German
3	31.10.2018	Threads	German
4	07.11.2018	Deadlocks	English
5	14.11.2018	Scheduling	English
6	21.11.2018	Races & Semaphores	German
7	28.11.2018	Memory Management 1	German
8	05.12.2018	Memory Management 2	German
9	12.12.2018	I/O	German
		Christmas Break	
10	16.01.2019	File Systems 1	German
11	23.01.2019	File Systems 2	German
12	30.01.2019	Virtualization	German
13	06.02.2019	OS Security	German
14	13.02.2019	Wrap Up	German

Exercises

Group	Weekday	CW*	Time	Room
1-A	Tuesday	Even	15:20 – 16:50	S1|01 A2 (karo 5)
1-B	Tuesday	Uneven	15:20 – 16:50	S1|01 A2 (karo 5)
2-A	Thursday	Even	11:40 – 13:10	S2|02 C110 (Piloty)
2-B	Thursday	Uneven	11:40 – 13:10	S2|02 C110 (Piloty)
3-A	Friday	Even	11:40 – 13:10	S1|01 A02 (karo 5)
3-B	Friday	Uneven	11:40 – 13:10	S1|01 A02 (karo 5)

*) CW = calendar week, see here.

- Group assignment via HRZ moodle (random for fairness)
- Please register before Oct 23rd (next Tue)
- Exercise sheets available Wednesday *before* the A week
- Solution (!) discussion in the exercise sessions
- Optional homework for self-assessment, feedback from tutors (submission via moodle)
- Office hour every Wednesday, 10:00-11:30, in S2|02 E203

Exams

- Midterm test exam
  - Test exam with *no "direct" influence on the course grade*
  - Wed, Dec 19th, 15:30 – 16:30, in S1|01 A1
  - Be there 15 minutes early!
  - Coverage: All we covered till then in lectures OR exercises
  - No registration on TUCaN!

- Final exam
  - Wed, Apr 10th, 10:00 – 12:00, rooms TBA
  - Be there 15 minutes early!
  - Coverage: All we covered till then in lectures OR exercises *including what has been covered in the midterm exam*
  - Registration on TUCaN required!

- Exam coverage: Lecture slides AND material covered elsewise in class

Course Contents

- Lecture
  - Fundamental OS concepts (Processes, Threads, Scheduling, Memory Management, …)
  - After that more advanced (& interesting) stuff (Security Concepts, Virtualization)

- Exercises
  - Active problem solving (the more active the more you gain!)

Relevant Literature + Lecture Foils

- ***Operating Systems: Three Easy Pieces***;
  R. & A. Arpaci-Dusseau, Available online:
  http://pages.cs.wisc.edu/~remzi/OSTEP/

- ***Modern Operating Systems***; A. Tanenbaum, Prentice Hall
- ***Operating System Concepts***; Silberschatz et al., John Wiley and Sons
- Both books available (in limited numbers) from the library
  - http://www.ulb.tu-darmstadt.de/
  - Bldg. S1|20, computer science section: 4th floor

- Slides will be made available via moodle
  - We will try to upload the slides before the lecture
  - Log in with your TU-ID and register for the course
  - Let us instantly know if you have trouble accessing the material!

OS History

- Initially there was no OS (just a single program)

OS History

- Initially there was no OS (just a single program)

- First Problem: Writing a second program for the same hardware
  - Re-implement input processing
  - Re-implement intermediate storage
  - Re-implement output processing

OS History

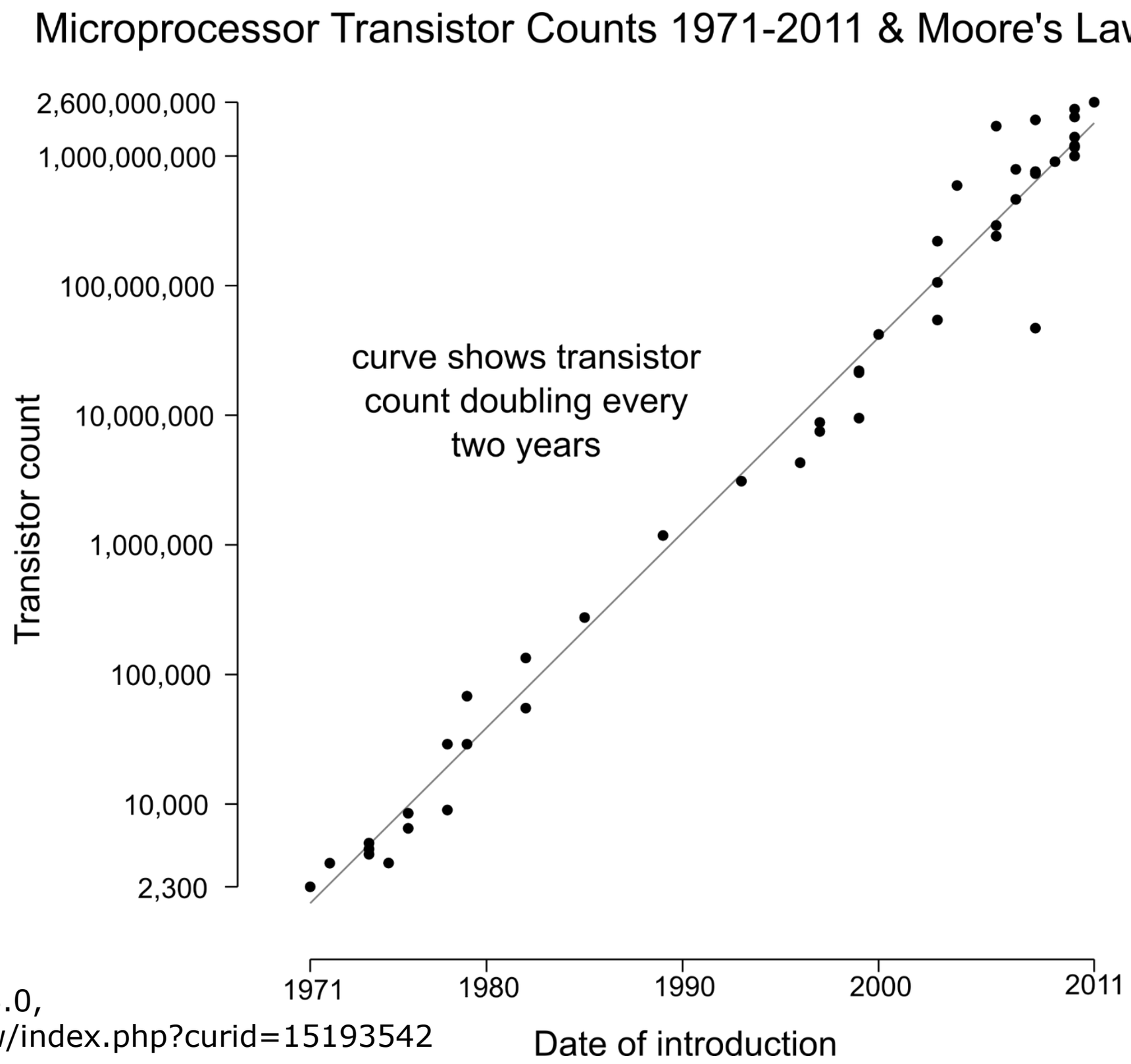
- Initially there was no OS (just a single program)

- First Problem: Writing a second program for the same hardware
  - Re-implement input processing
  - Re-implement intermediate storage
  - Re-implement output processing

- S
  - Re-implement input processing
  - …

OS History

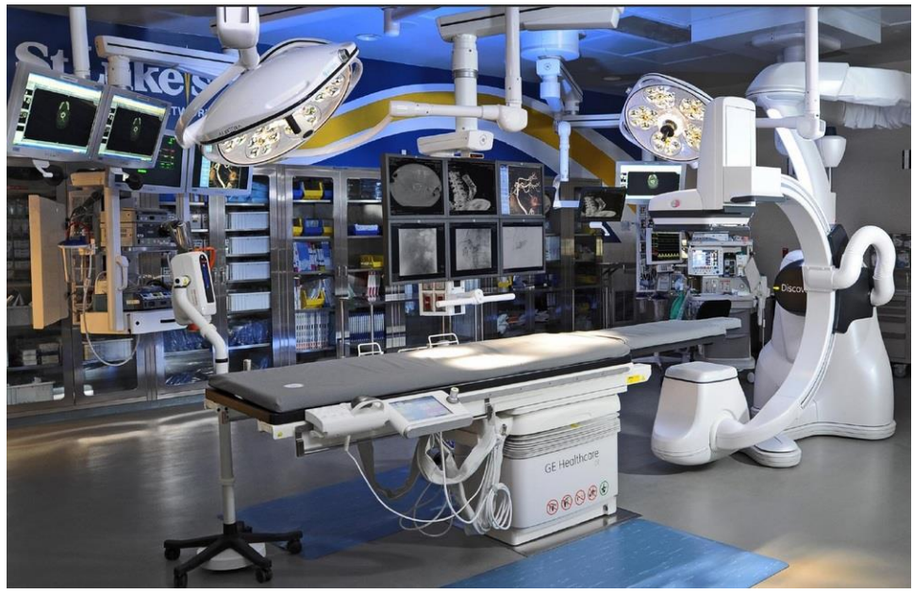
- Third problem: Increasing pace of change



Based on a plot by
Wgsimon - Own work, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=15193542

OSs Today

- Now:
  - distributed, networked, virtual machines, multi-user, multi-core, …
  - running everywhere (on mobile phones, tablets, mainframes, data centers, etc.)
  - commercial-off-the-shelf (COTS) products



OSs Today

- Now:
  - distributed, networked, virtual machines, multi-user, multi-core, …
  - running everywhere (on mobile phones, tablets, mainframes, data centers, etc.)
  - commercial-off-the-shelf (COTS) products

- First problem (rewriting code for 2nd prog on same HW)
  - Solved

OSs Today

- Now:
  - distributed, networked, virtual machines, multi-user, multi-core, …
  - running everywhere (on mobile phones, tablets, mainframes, data centers, etc.)
  - commercial-off-the-shelf (COTS) products

- First problem (rewriting code for 2nd prog on same HW)
  - Solved

- Second & third problem (rewriting code for new HW)
  - Shifted to updating the OS or its hardware abstraction layer (HAL)

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)

OS Definition 1: Hardware Abstraction

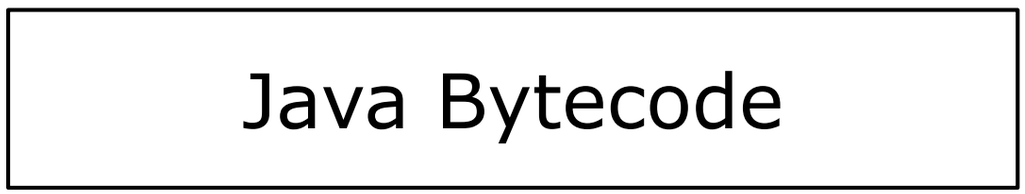
OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

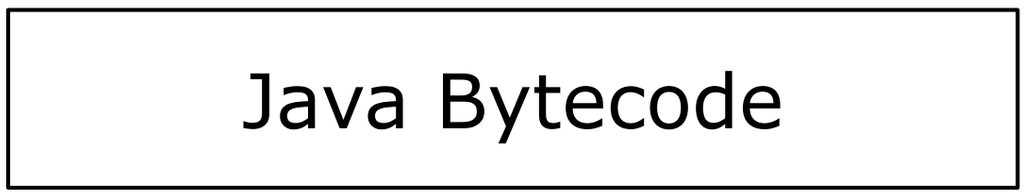
- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

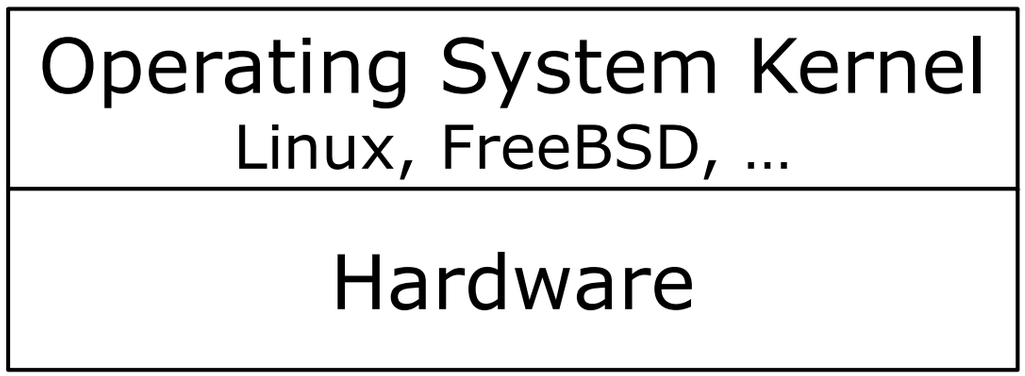


OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

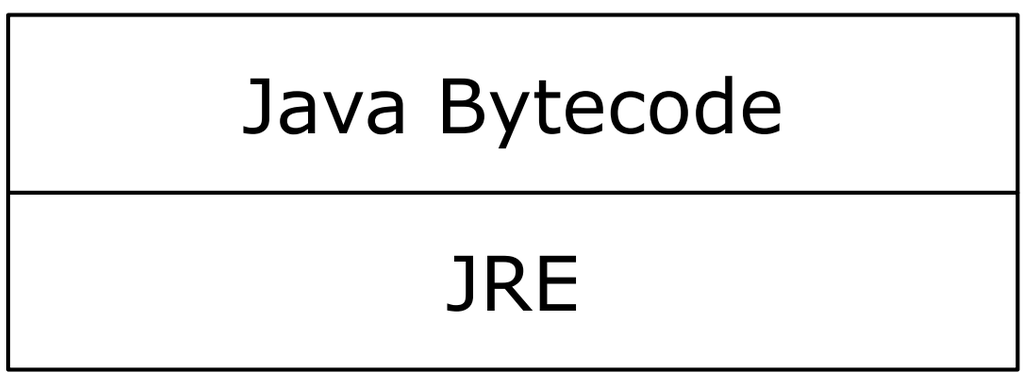


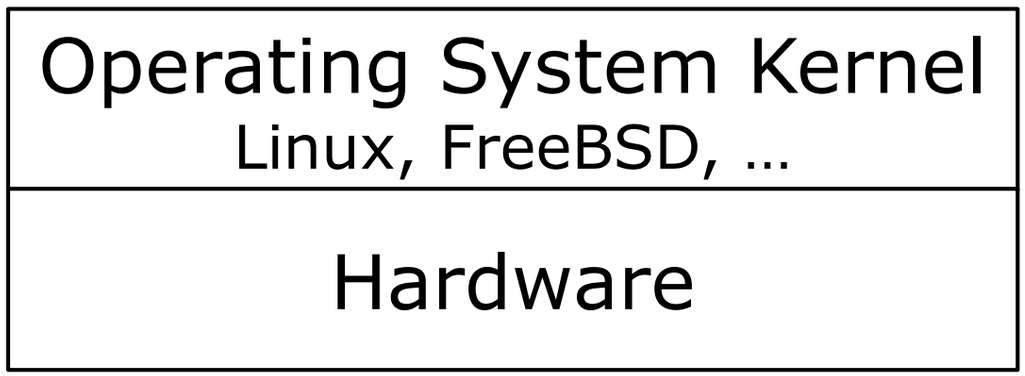


OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

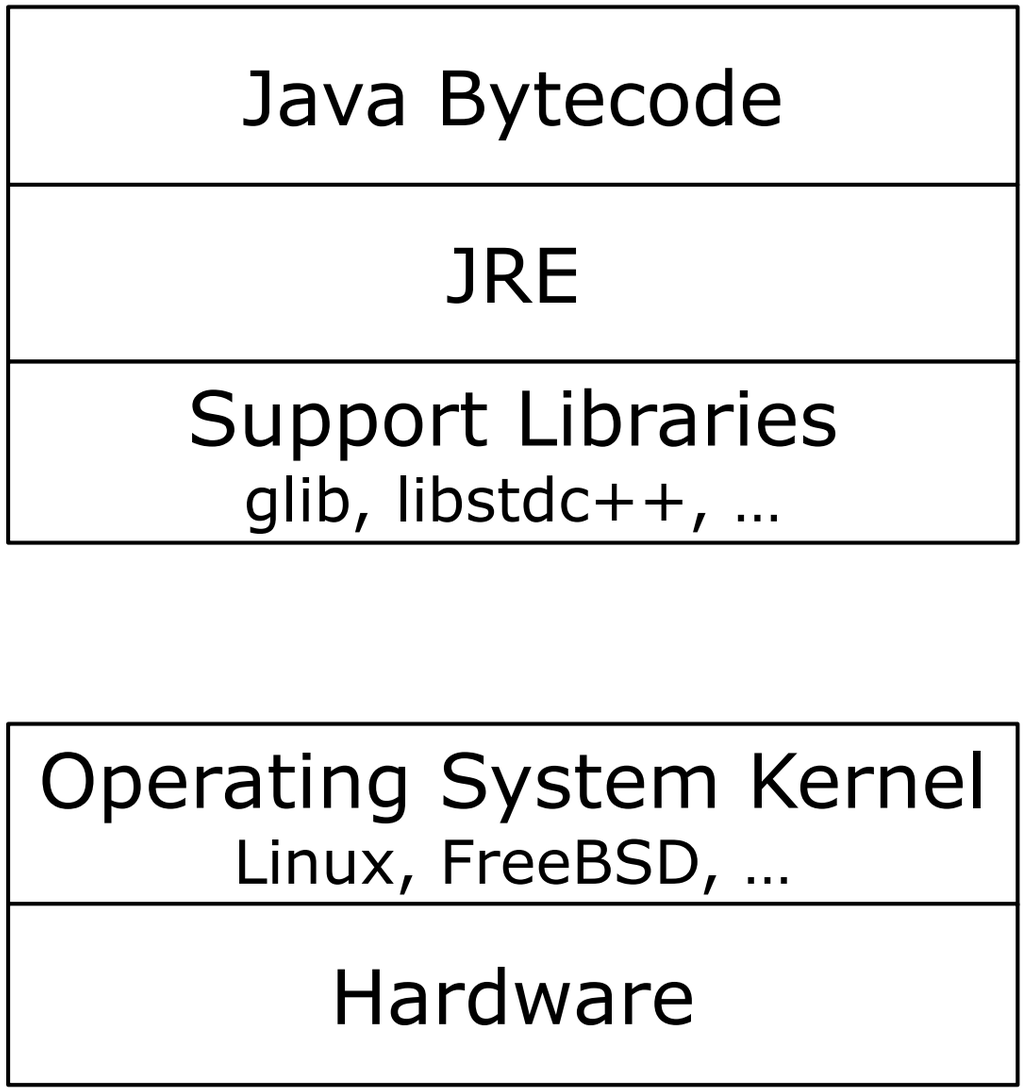




OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

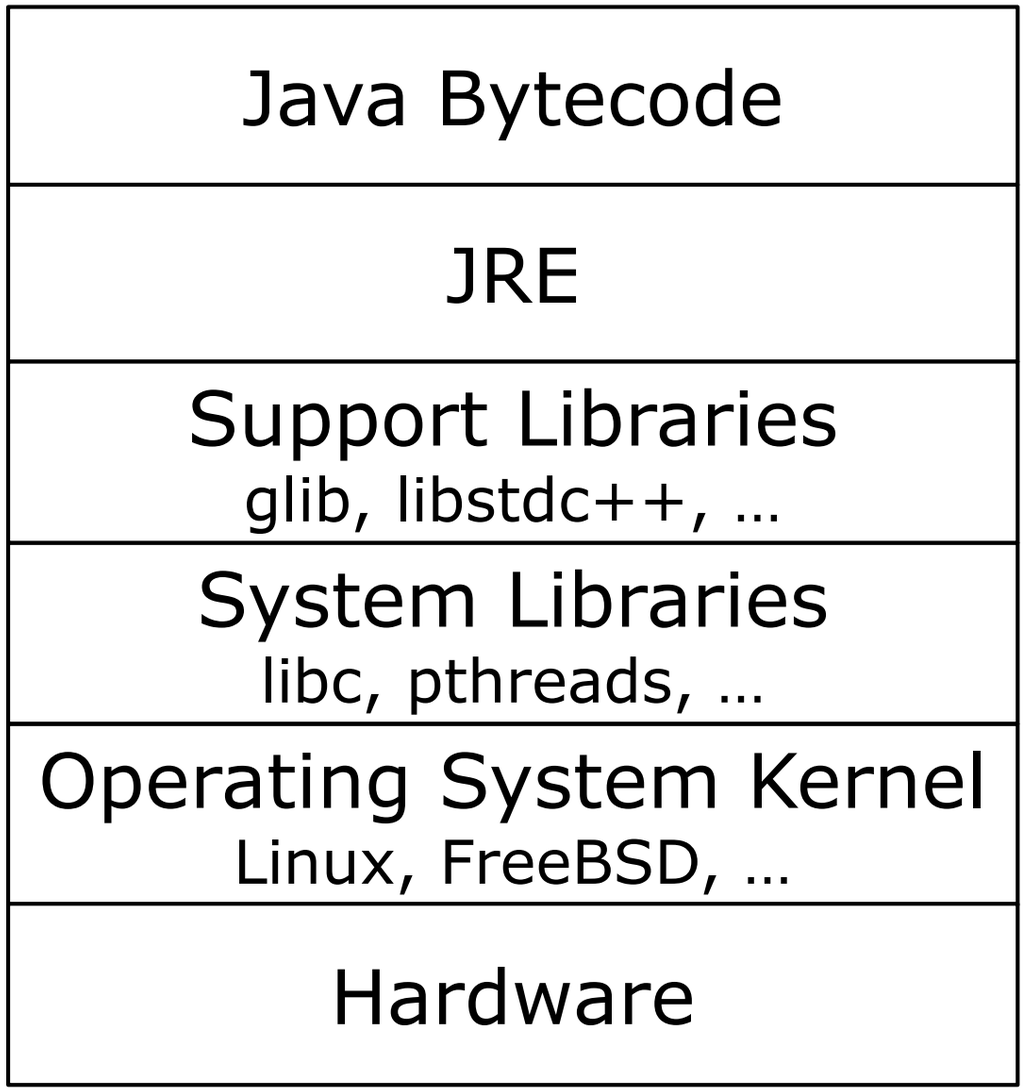
- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

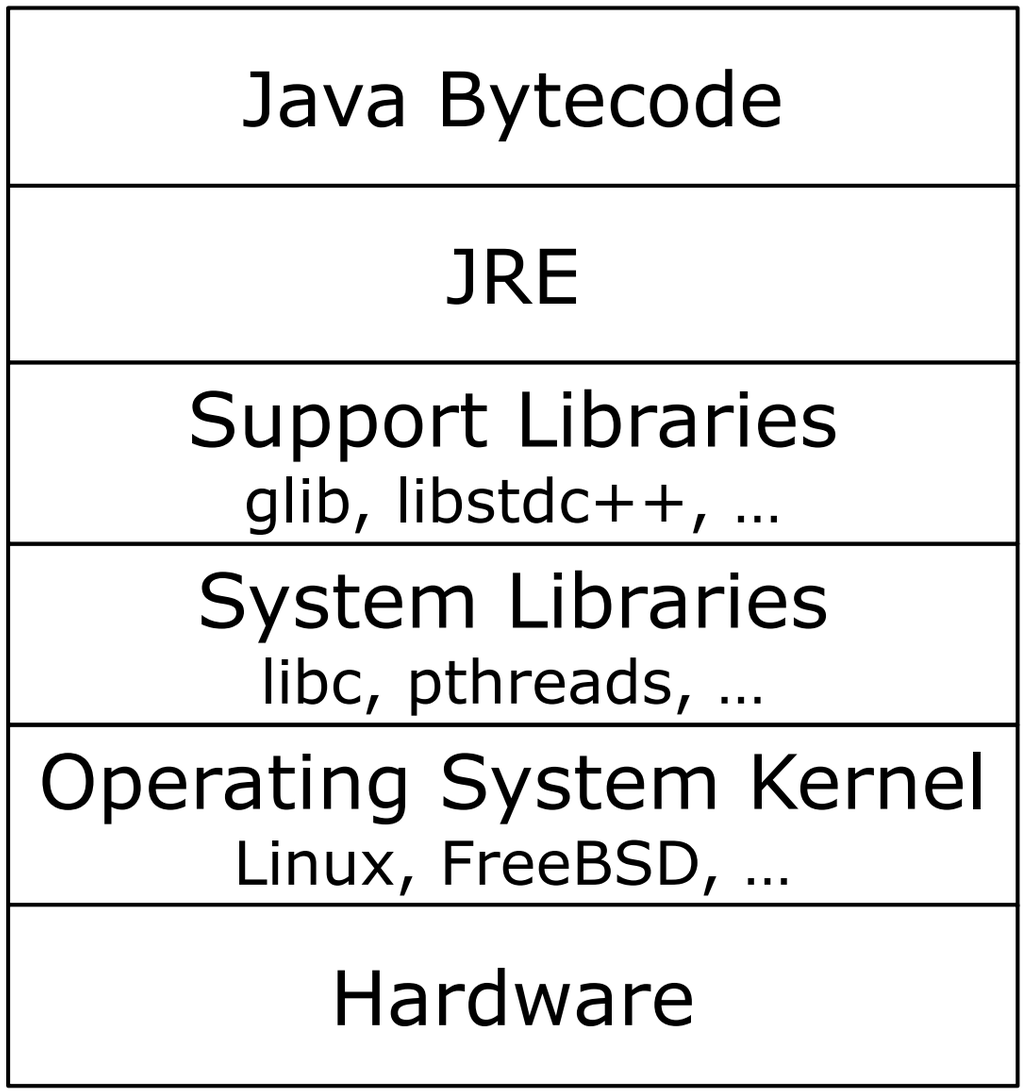
- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

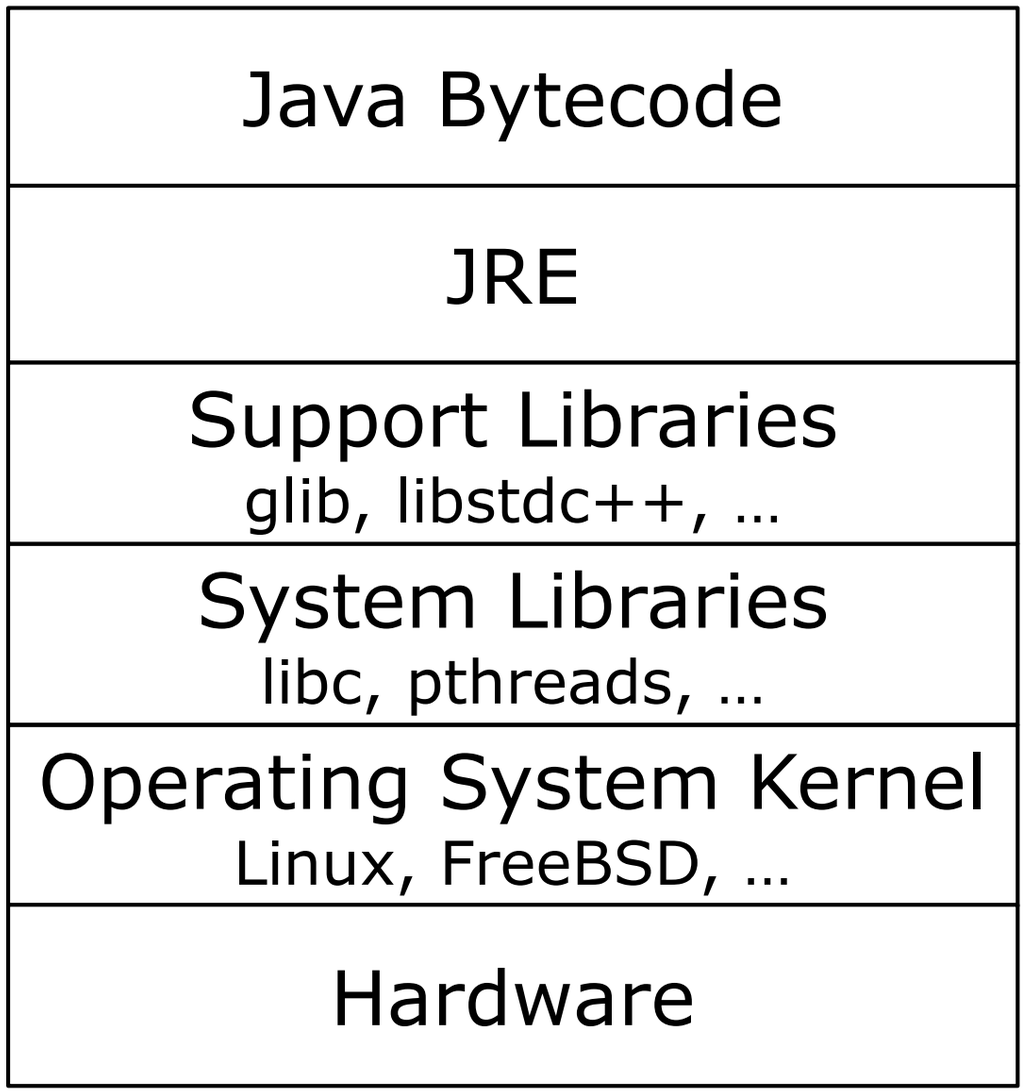


a few KLOC?

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



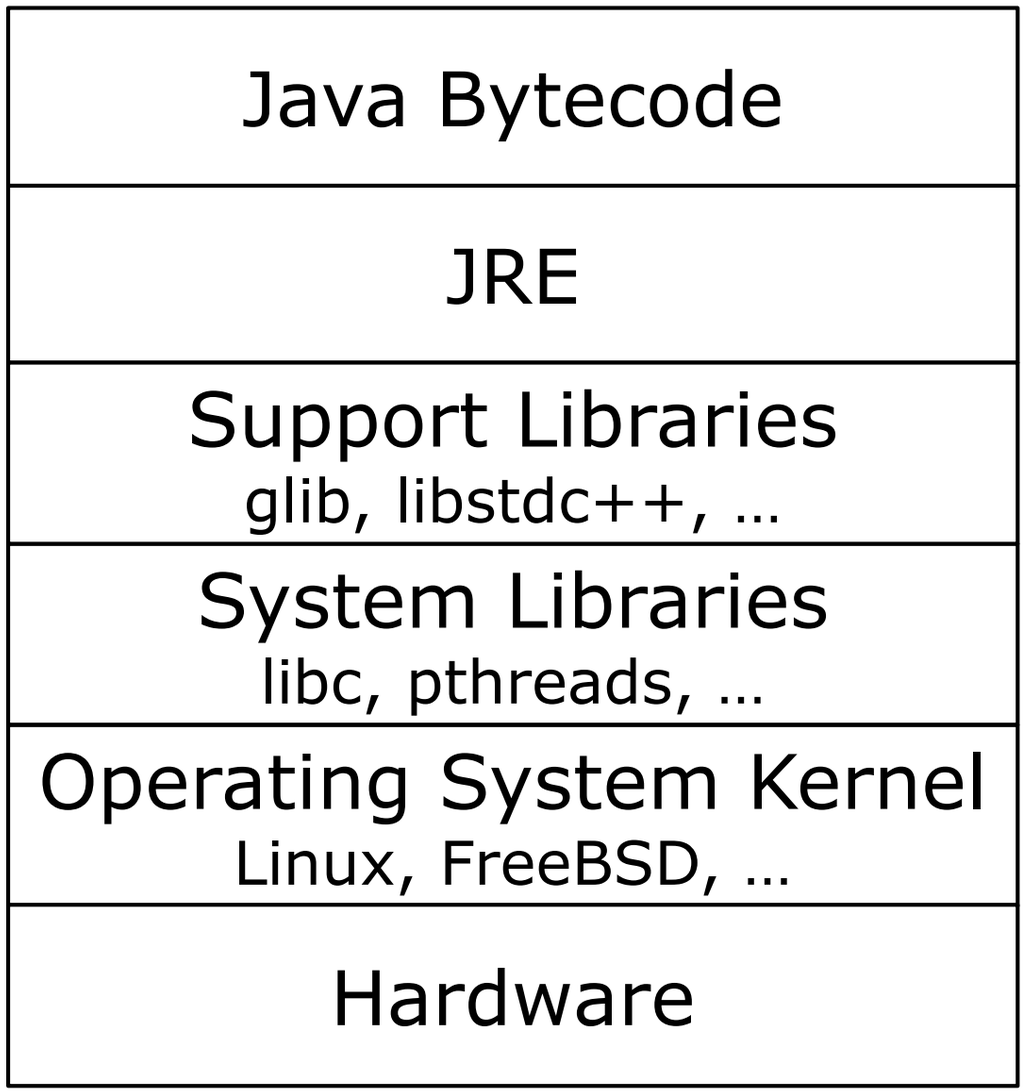
a few KLOC?

~ 4.5 MLOC (openjdk8)

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



a few KLOC?

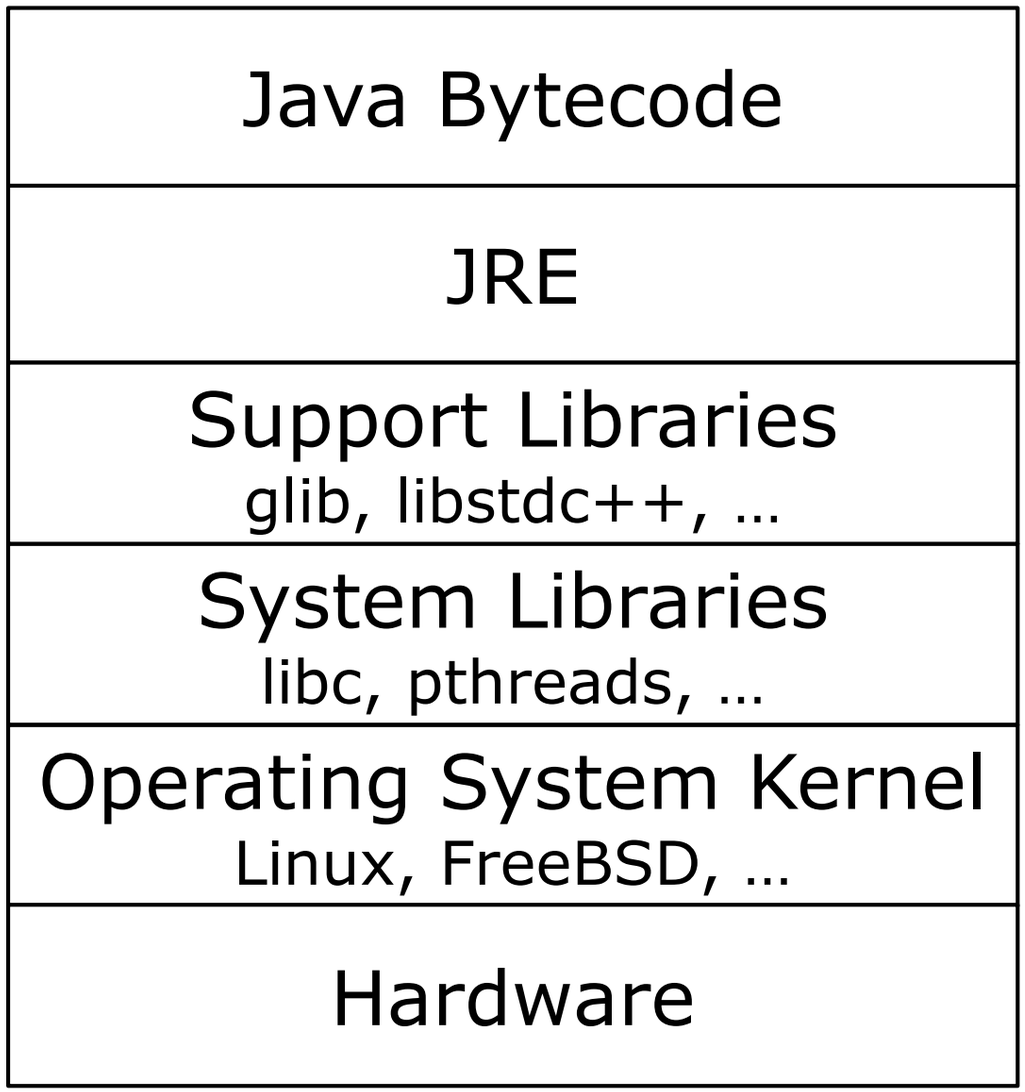
~ 4.5 MLOC (openjdk8)

several MLOC

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



a few KLOC?

~ 4.5 MLOC (openjdk8)

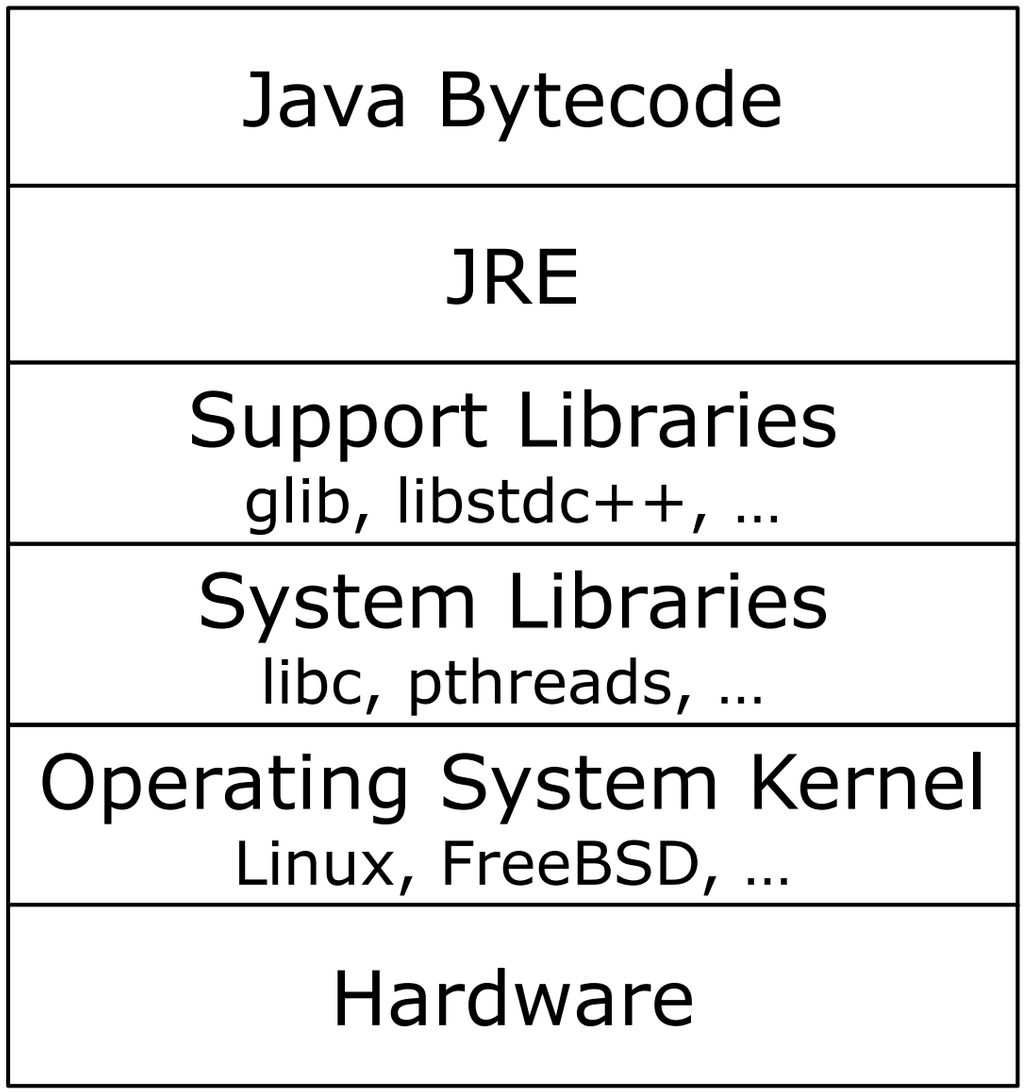
several MLOC

1-2 MLOC

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



a few KLOC?

~ 4.5 MLOC (openjdk8)

several MLOC

1-2 MLOC

15-25 MLOC

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

Java Bytecode	a few KLOC?
JRE	~ 4.5 MLOC (openjdk8)
Support Libraries glib, libstdc++, …	several MLOC
System Libraries libc, pthreads, …	1-2 MLOC
Operating System Kernel Linux, FreeBSD, …	15-25 MLOC
Hardware	

"Operating System" in this course

OS Definition 1: Hardware Abstraction

OS: program (or set of programs) providing

- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*

Java Bytecode	a few KLOC?
JRE	~ 4.5 MLOC (openjdk8)
Support Libraries glib, libstdc++, …	several MLOC
System Libraries libc, pthreads, …	1-2 MLOC
Operating System Kernel Linux, FreeBSD, …	15-25 MLOC
Hardware	

"Operating System" in this course

Alternative definition: "everything that runs in kernel mode"

The Amount of Code You Blindly Trust

Windows Version	~ SLOC
Windows NT 4.0	11-12M
Windows 2000	29+M
Windows XP	40M
Windows Server 2003	50M

The Amount of Code You Blindly Trust

Windows Version	~ SLOC
Windows NT 4.0	11-12M
Windows 2000	29+M
Windows XP	40M
Windows Server 2003	50M

Debian Unstable	~ SLOC
2001	1.1M
2005	4.0M
2009	26.3M
2013	72.0M
2017	86.4M

The Amount of Code You Blindly Trust

Windows Version	~ SLOC
Windows NT 4.0	11-12M
Windows 2000	29+M
Windows XP	40M
Windows Server 2003	50M

Debian Unstable	~ SLOC
2001	1.1M
2005	4.0M
2009	26.3M
2013	72.0M
2017	86.4M

Linux Kernel	~ SLOC
2005	4.7M
2009	8.2M
2013	12M
2017	16.8M

OS Definition 1: Hardware Abstraction

- Definition 1.1: Kernel plus system libraries
  - Problem: What's a "system library"?
  - Solution: Standards (POSIX, LSB, WinAPI, …)

- Definition 1.2: Kernel (i.e., everything that runs in kernel mode)
  - Problem 1: Needs definition what kernel mode is
  - Problem 2: Does not apply for hardware architectures that do not follow this definition
  - Solution: Restriction to certain hardware architectures

OS Definition 1.1 Example: POSIX

- Portable Operating System Interface
- Maintained by Austin Group
- Specifies the OS interface, e.g., functions to open files, etc.
- IEEE Std 1003.x & ISO/IEC 9945
- First publication 1988
- Current edition (2016) available online: http://pubs.opengroup.org/onlinepubs/9699919799/

- API standard: Write once, compile everywhere

- POSIX compliance certification available through IEEE & The Open Group

OS Definition 1.1 Example: LSB

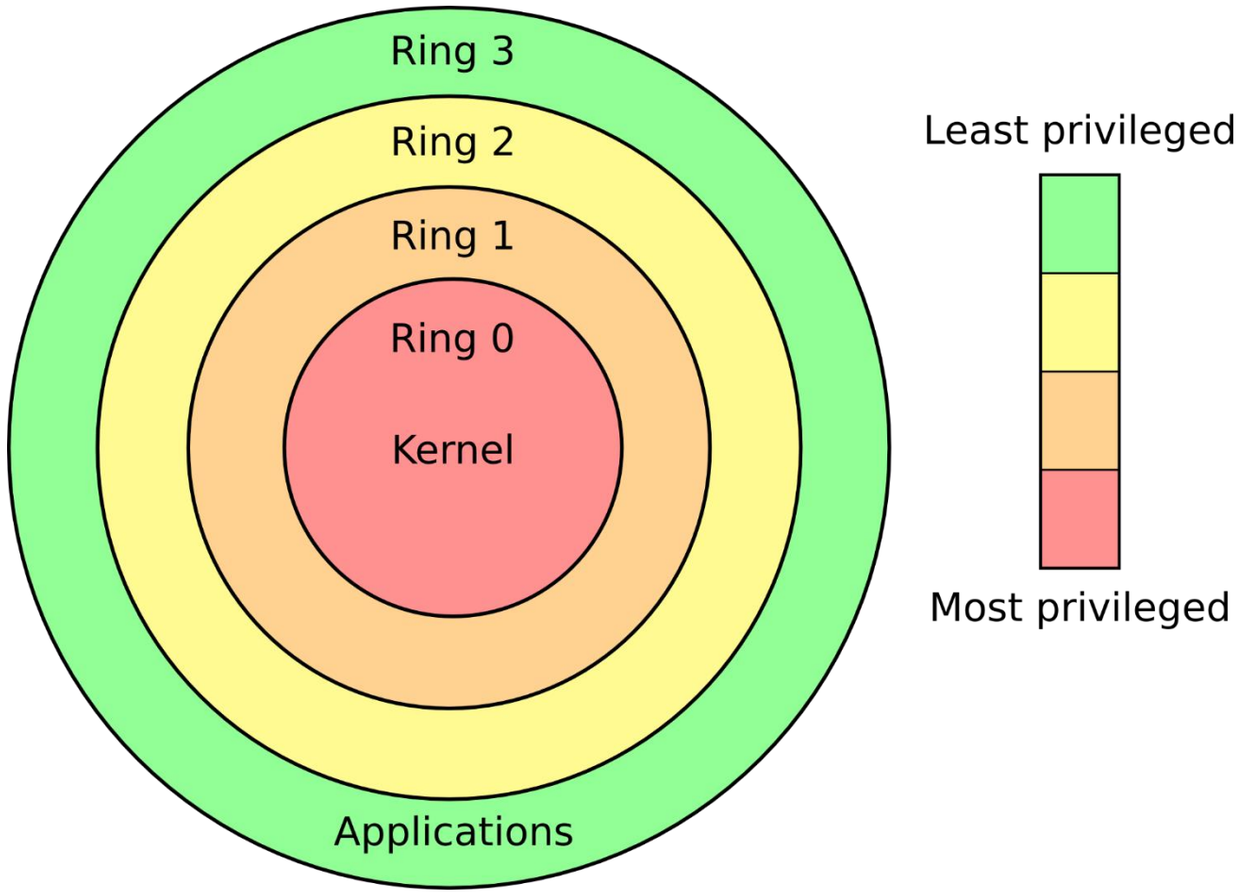
- The Linux Standard Base specifies Linux's interface
- Maintained by the Linux Foundation
- Large overlap with POSIX
- First release 2001
- Version 3.1 (2006) has been registered as ISO standard ISO/IEC 23360
- Latest release (and all prior) available online:
  http://refspecs.linuxfoundation.org/lsb.shtml
  with plenty additional resources:
  https://www.linuxbase.org/download/

- ABI standard: Compile once, run everywhere

- LSB compliance certification through the Linux Foundation

OS Definition 1.2 Example: x86 Rings

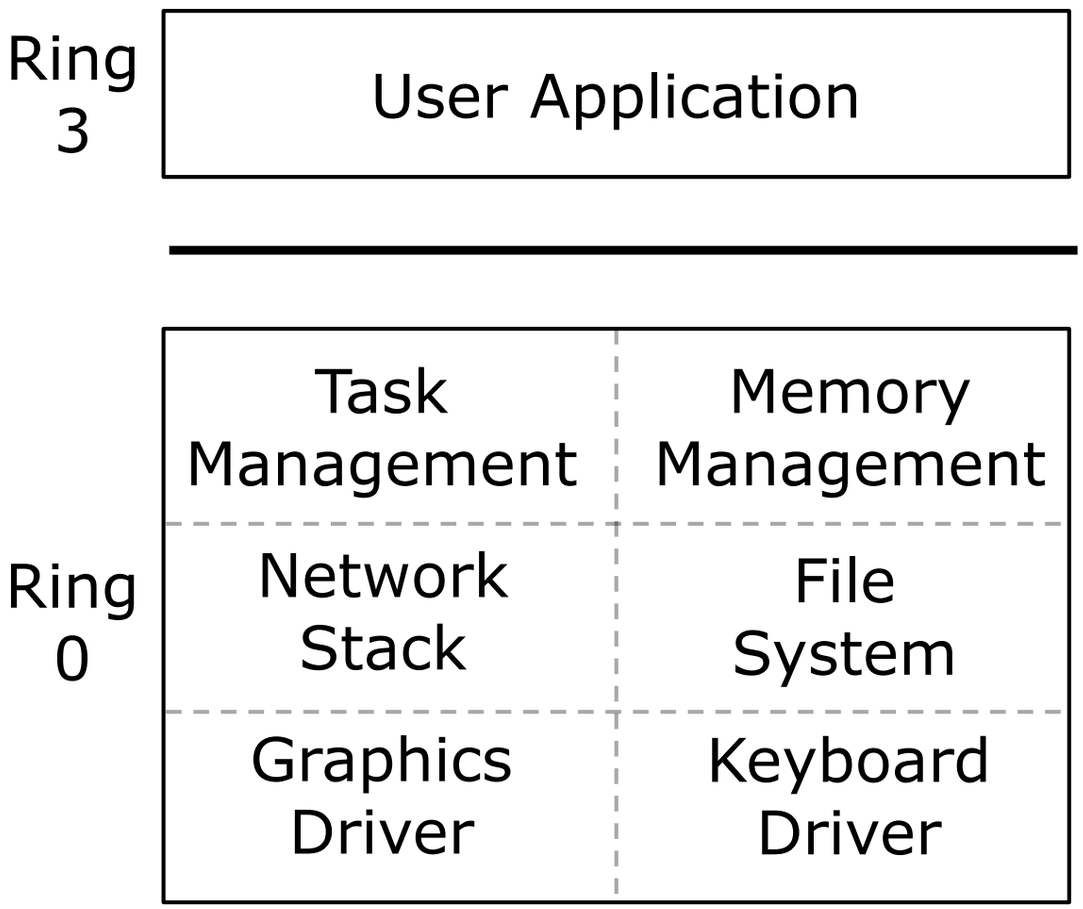
- Intel x86 defines multiple protection "rings"
- Some "privileged" instructions can only be executed in in certain rings
- For example, to modify which memory is visible to which program, a ring 0 instruction is needed
- To prevent any program from disrupting the OS, only the OS can execute in ring 0 (a.k.a. "**kernel mode**" vs. "**user mode**" in ring 3)
- Problem: Where to draw the boundary what to execute in ring 0 and what not?



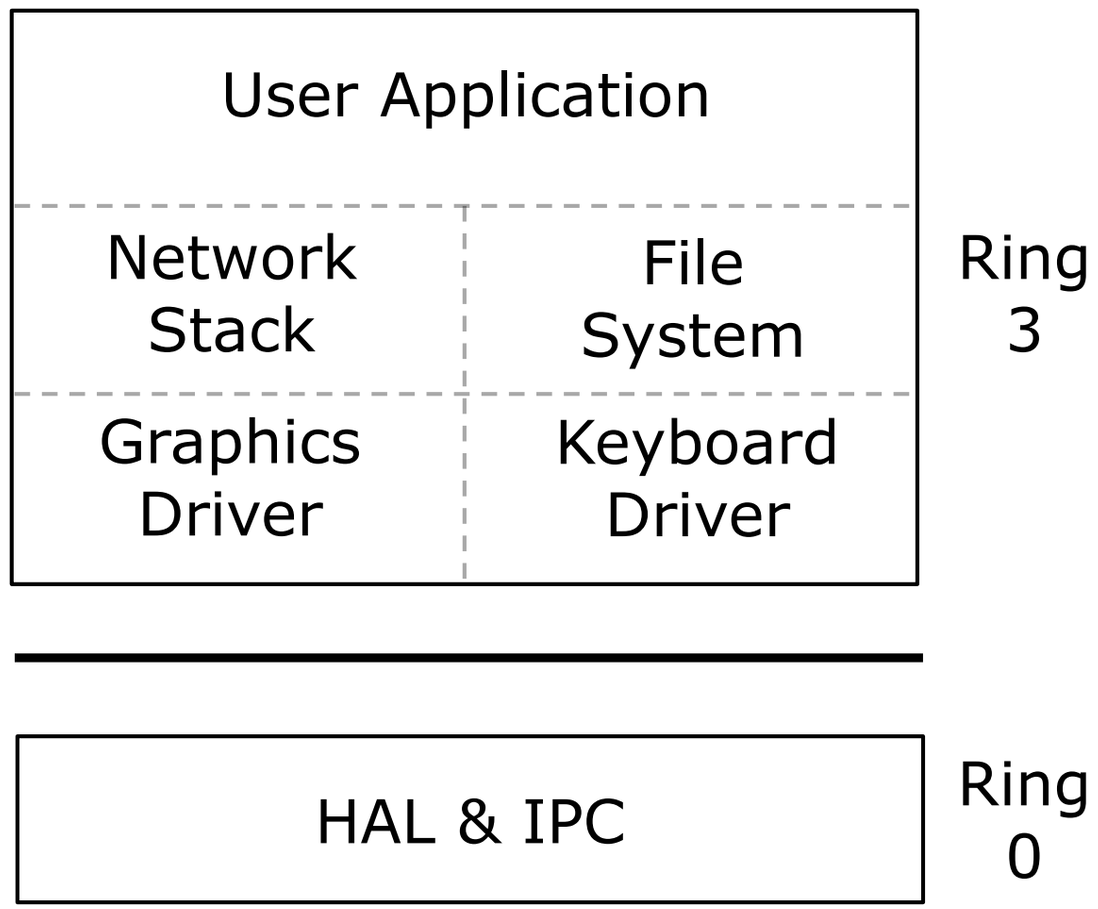
Adapted from an illustration by Hertzsprung at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=8950144

OS Definition 1.2: Monolithic vs Microkernel

Monolithic OS Architecture



Microkernel Architecture



OS Definition 2: Coordination

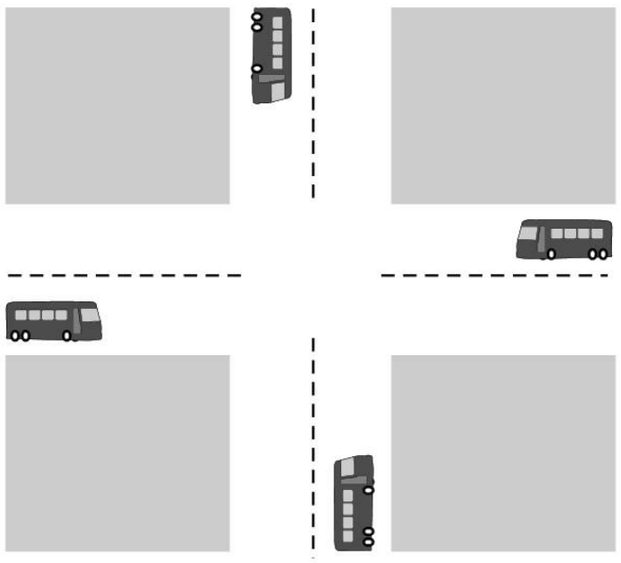
- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its share of the resource(s)

OS Definition 2: Coordination

- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its share of the resource(s)
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the resources (e.g., HW or data structures) – ordering, sequencing, …

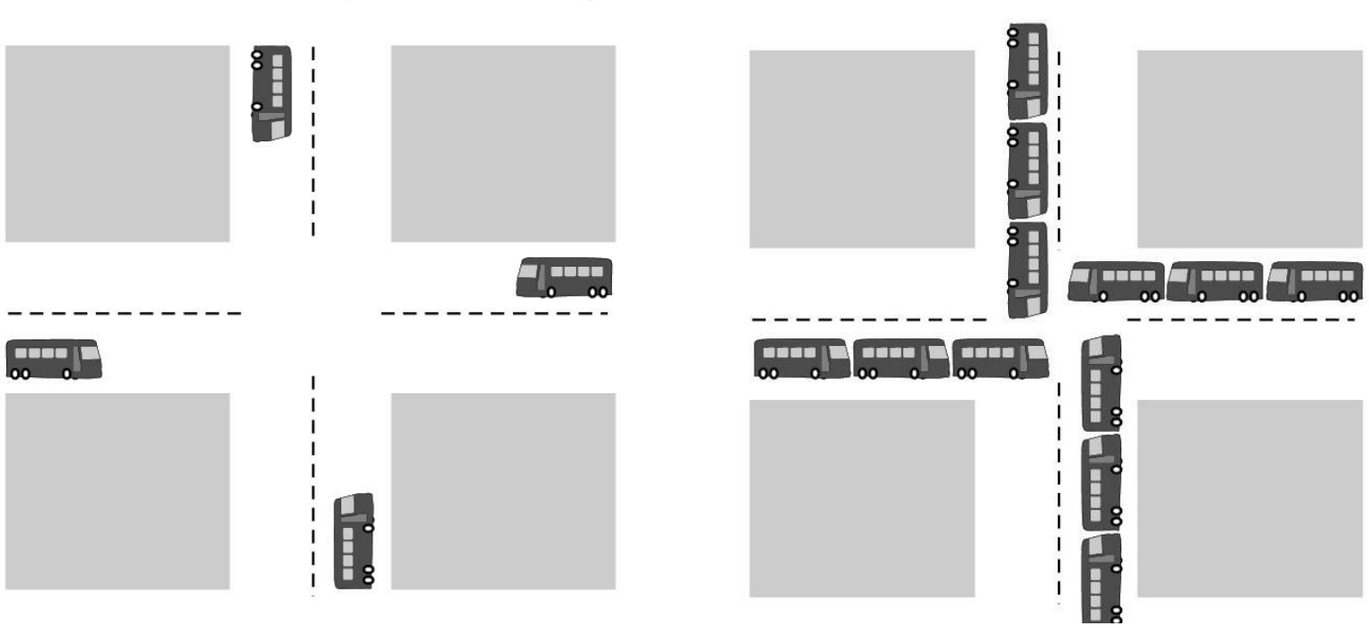
OS Definition 2: Coordination

- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its share of the resource(s)
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the resources (e.g., HW or data structures) – ordering, sequencing, …



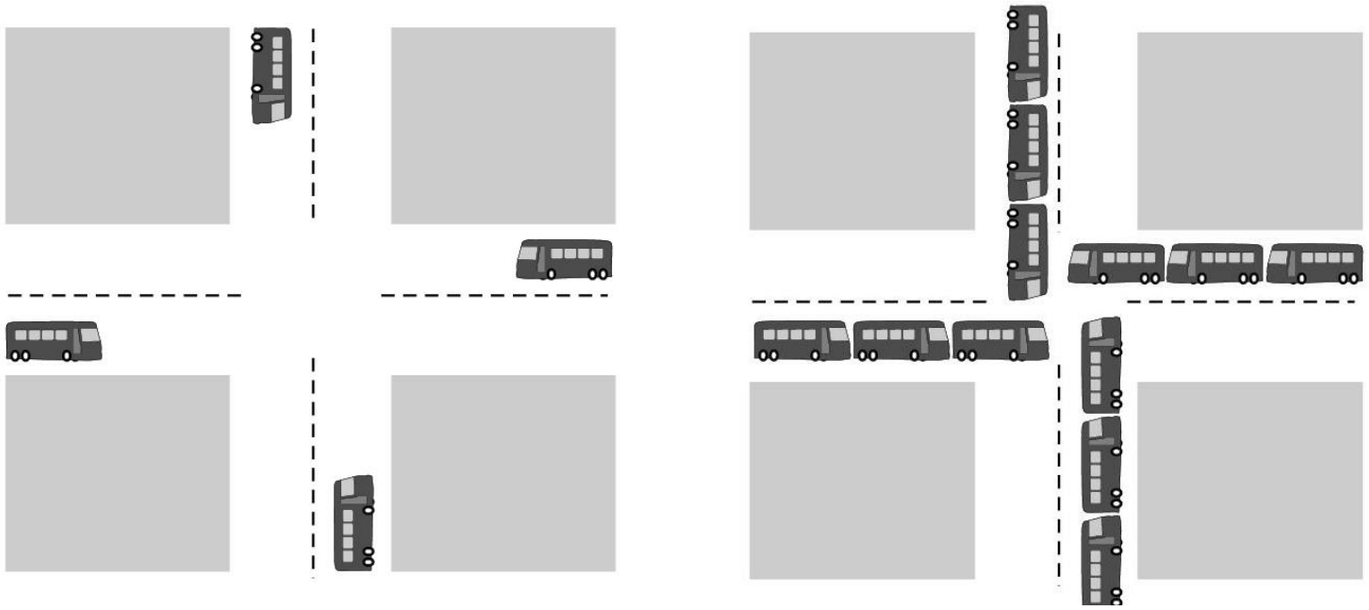
OS Definition 2: Coordination

- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its share of the resource(s)
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the resources (e.g., HW or data structures) – ordering, sequencing, …



OS Definition 2: Coordination

- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its share of the resource(s)
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the resources (e.g., HW or data structures) – ordering, sequencing, …



Upper images © A.Tanenbaum, "Modern Operating Systems".
Right image © A.G. Stumpf, "Traffic",
https://creativecommons.org/licenses/by/2.0/

Course Outline

Basic Topics

- Hardware Abstraction
  - Processes, Threads, and IPC
  - Memory Management
  - I/O
- Resource Management and Control
  - Concurrency
  - Resource Allocation/Scheduling
  - Deadlocks, Race Conditions & Mutual Exclusion
- Persistence: File Systems

Advanced Topics

- Virtualization
- OS Security