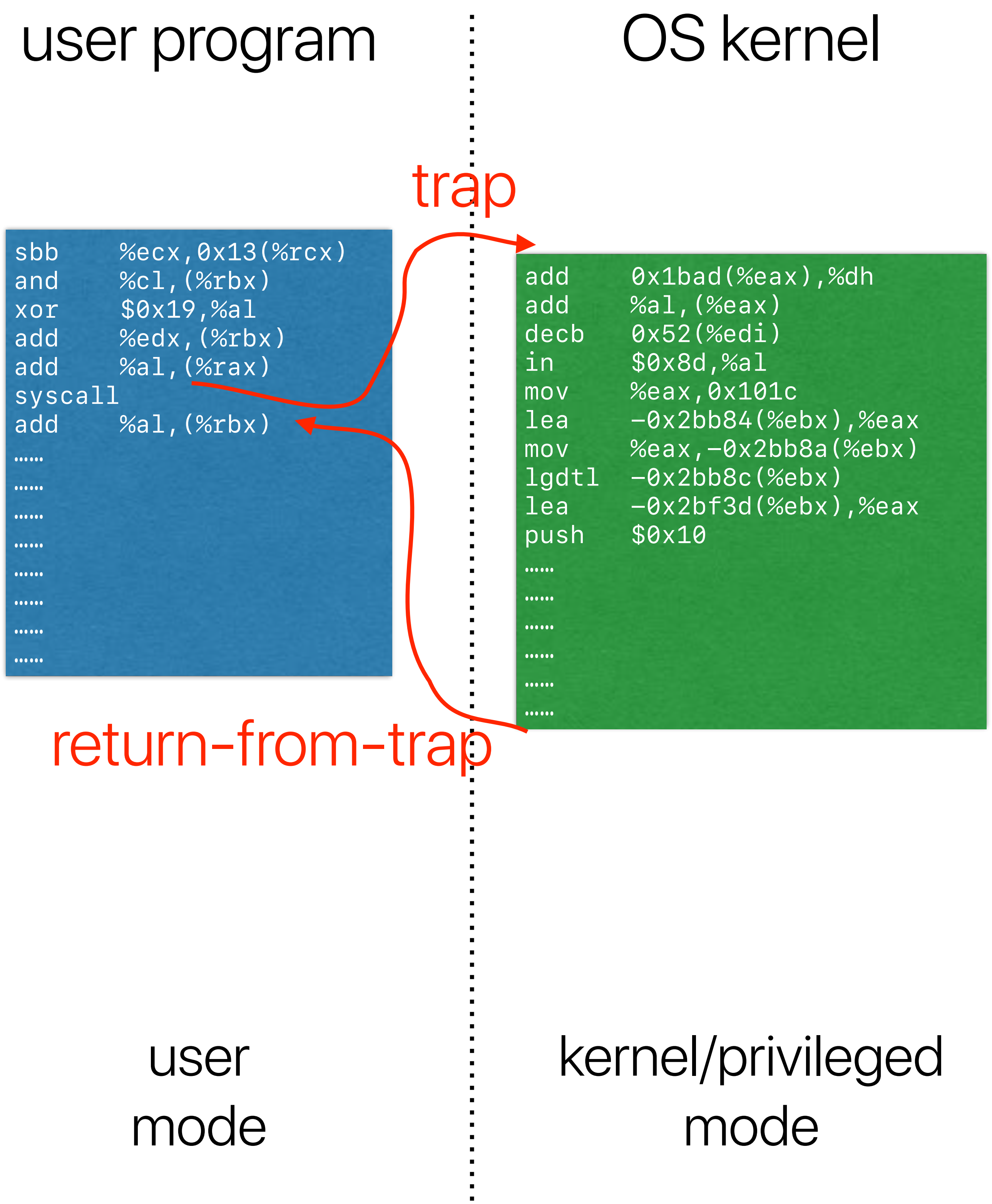
Design philosophy of operating systems (II)

Hung-Wei Tseng

Recap: How applications can use privileged operations?

- Implemented in “trap” instructions
  - Raise an exception in the processor
  - The processor saves the exception PC and jumps to the corresponding exception handler in the OS kernel
- The OS kernel only get involved when necessary
  - System calls
  - Hardware interrupts
  - Exceptions
- The OS kernel works **on behave of** the requesting process — not a process
  - Somehow like a function call to a dynamic linking library
  - As a result — overhead of copying registers, allocating local variables for kernel code and etc…



```
sbb     %ecx,0x13(%rcx)
and     %cl,(%rbx)
xor     $0x19,%al
add     %edx,(%rbx)
add     %al,(%rax)
syscall
add     %al,(%rbx)
……
……
……
……
……
……
……
……
```

```
add     0x1bad(%eax),%dh
add     %al,(%eax)
decb    0x52(%edi)
in      $0x8d,%al
mov     %eax,0x101c
lea     -0x2bb84(%ebx),%eax
mov     %eax,-0x2bb8a(%ebx)
lgdtl   -0x2bb8c(%ebx)
lea     -0x2bf3d(%ebx),%eax
push    $0x10
……
……
……
……
……
……
```

Recap: THE

- Why should people care about this paper in 1968?
  - Turn-around time of **short** programs
  - Economic use of peripherals
  - Automatic control of backing storage
  - Economic use of the machine

Process Abstraction

Virtual memory

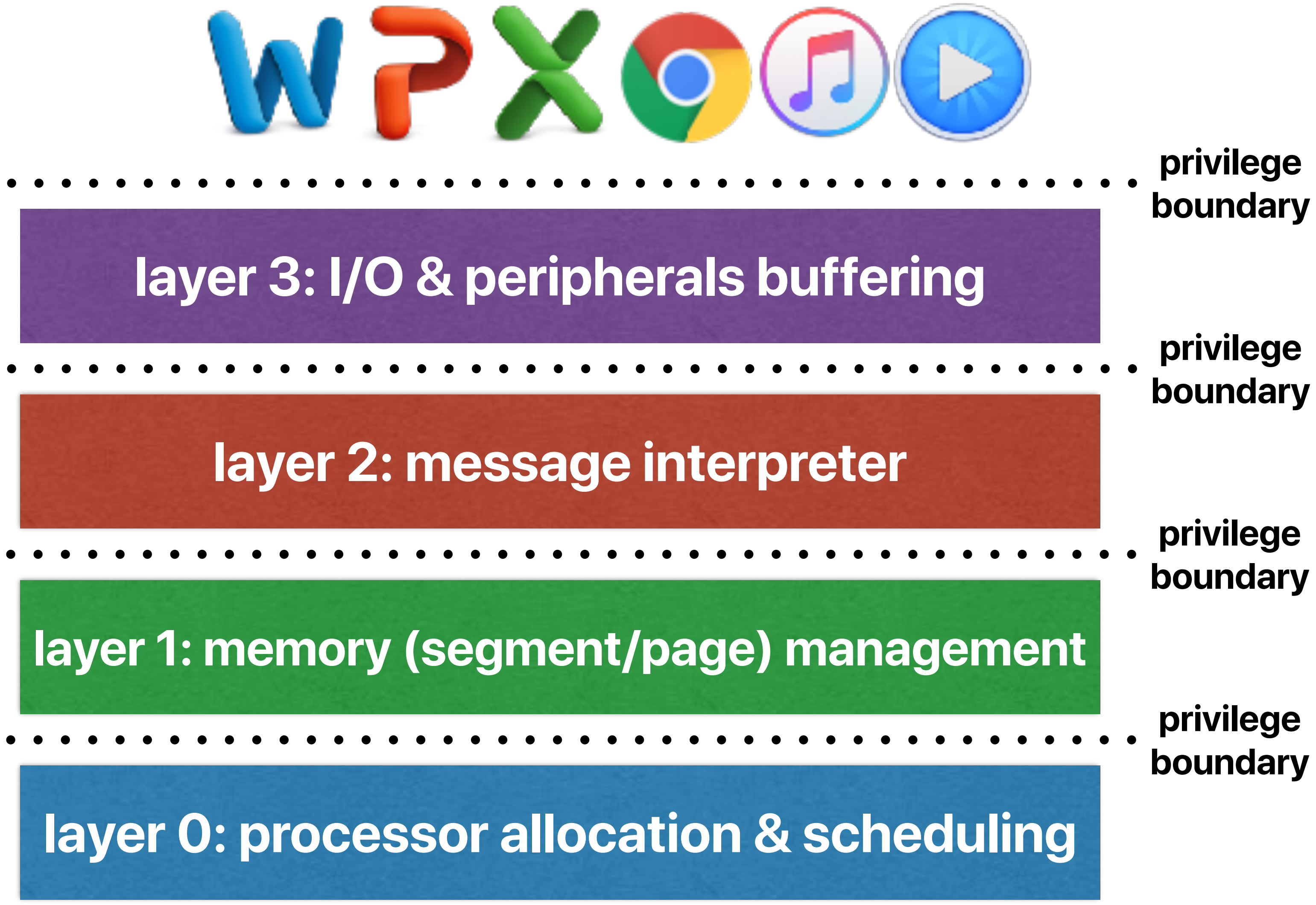
Mutex

- Designing a system is difficult in 1968
  - Difficult to verify soundness
  - Difficult to prove correctness
  - Difficult to deal with the complexities

Layered Design

Recap: THE

THE



privilege boundary

layer 3: I/O & peripherals buffering

privilege boundary

layer 2: message interpreter

privilege boundary

layer 1: memory (segment/page) management

privilege boundary

layer 0: processor allocation & scheduling

The overhead of kernel switches/system calls

- On a 3.7GHz intel Core i5-9600K Processor, please make a guess of the overhead of switching from user-mode to kernel mode.

  A. a single digit of nanoseconds

  B. tens of nanoseconds

  [✓ C. hundreds of nanoseconds]

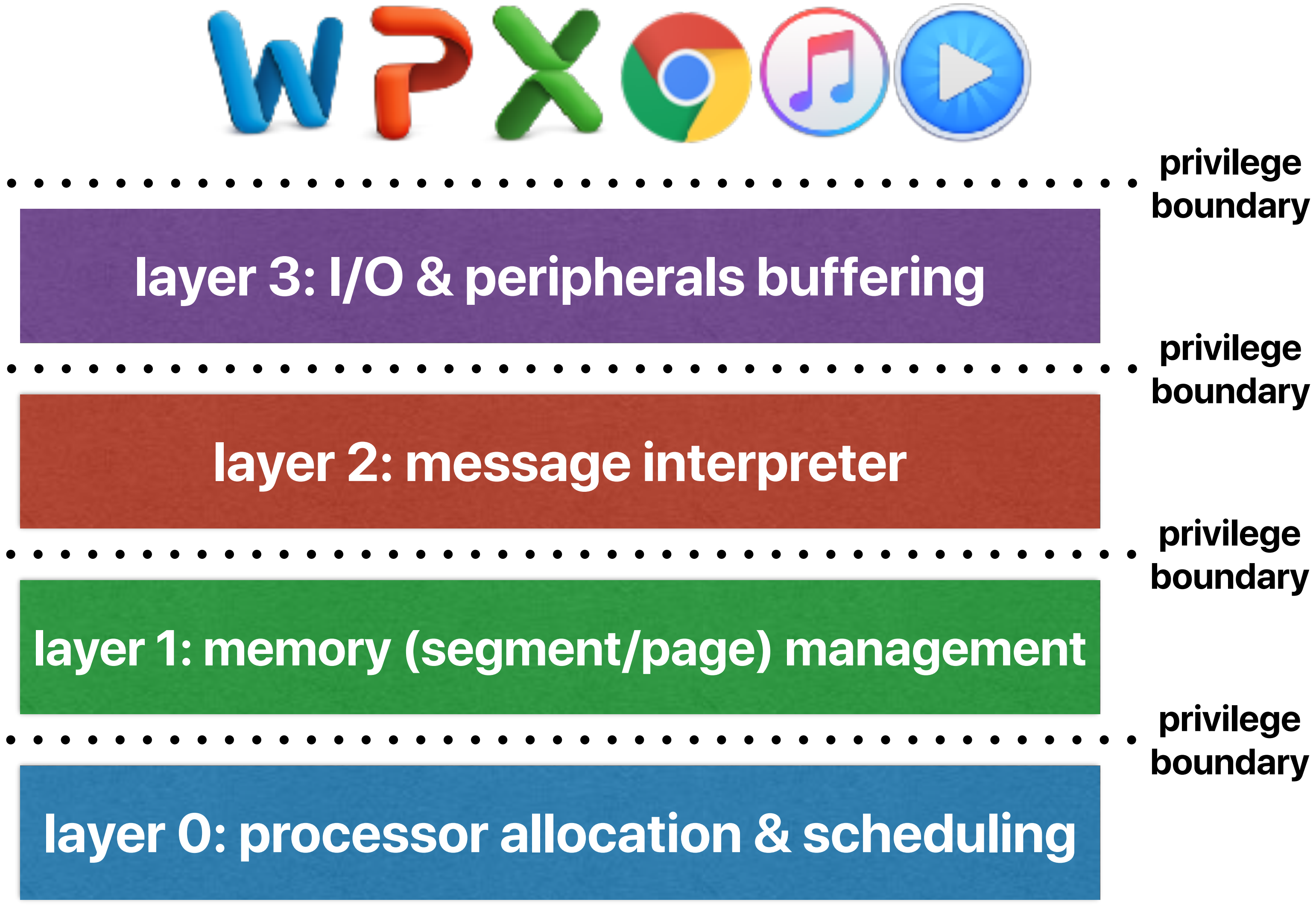
  D. a single digit of microseconds

  E. tens of microseconds

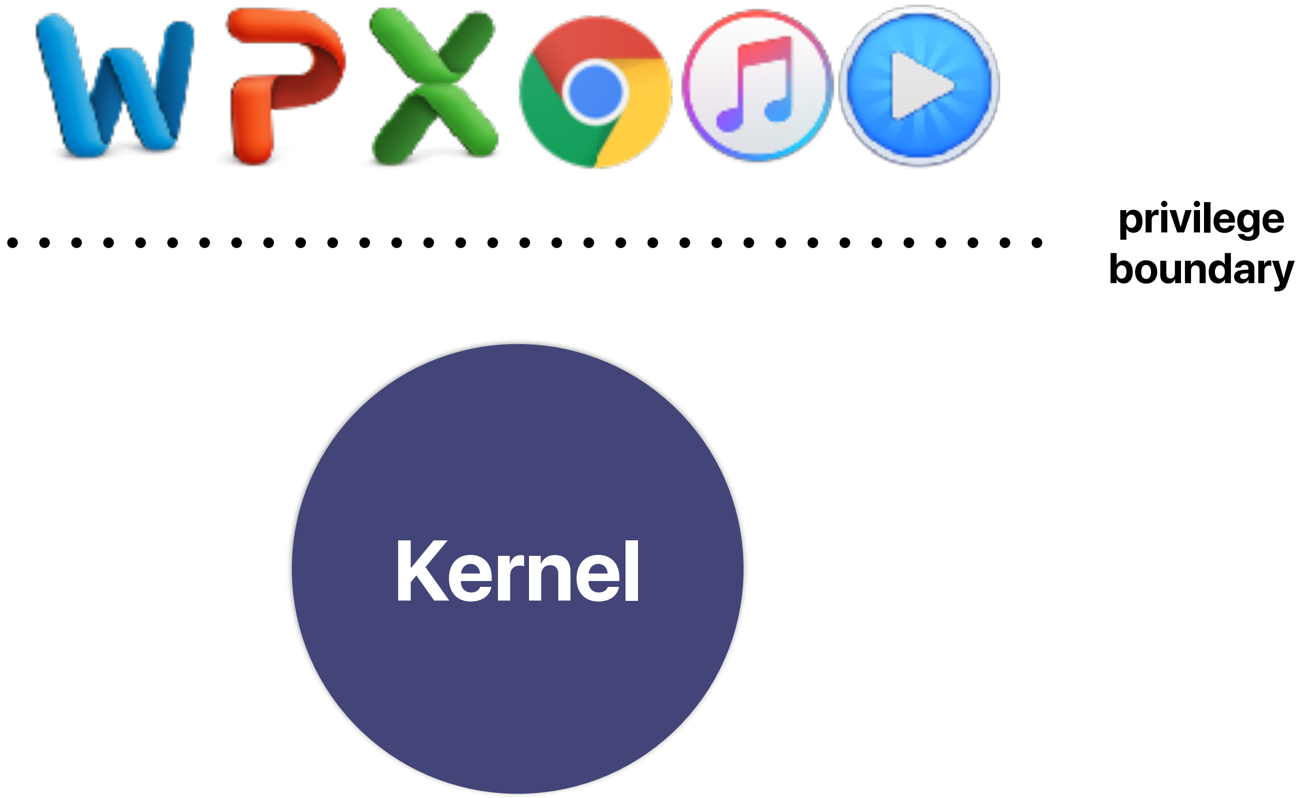
Operations	Latency (ns)
L1 cache reference	1 ns
Branch mispredict	3 ns
L2 cache reference	4 ns
Mutex lock/unlock	17 ns
Send 2K bytes over network	44 ns
Main memory reference	100 ns
Read 1 MB sequentially from memory	3,000 ns
Compress 1K bytes with Zippy	2,000 ns
Read 4K randomly from SSD*	16,000 ns
Read 1 MB sequentially from SSD*	49,000 ns
Round trip within same datacenter	500,000 ns
Read 1 MB sequentially from disk	825,000 ns
Disk seek	2,000,000 ns
Send packet CA-Netherlands-CA	150,000,000 ns

Recap: THE v.s. Hydra

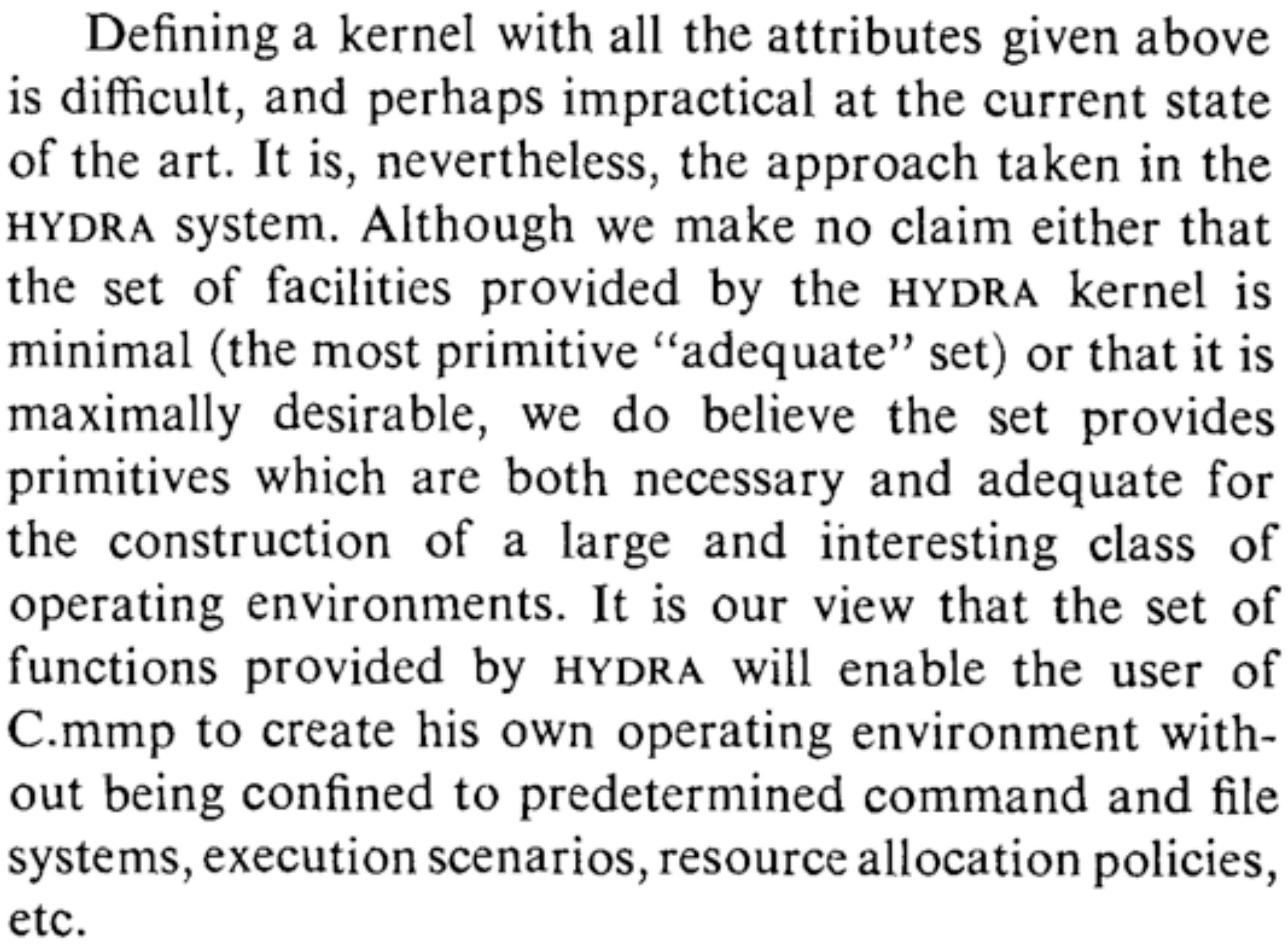
THE


privilege boundary
layer 3: I/O & peripherals buffering
privilege boundary
layer 2: message interpreter
privilege boundary
layer 1: memory (segment/page) management
privilege boundary
layer 0: processor allocation & scheduling

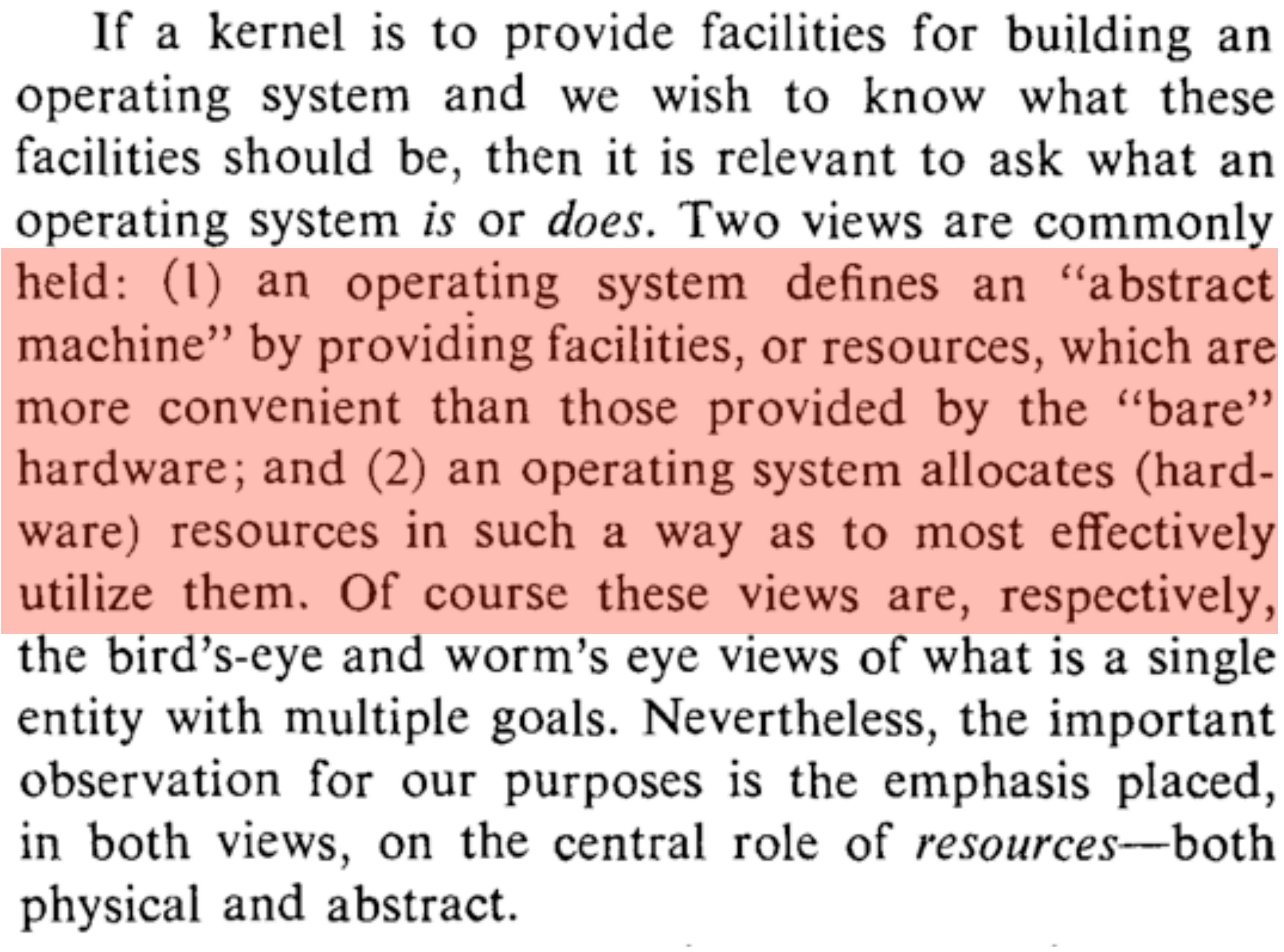
Hydra



Recap: the concept "Kernel" in Hydra



Defining a kernel with all the attributes given above is difficult, and perhaps impractical at the current state of the art. It is, nevertheless, the approach taken in the HYDRA system. Although we make no claim either that the set of facilities provided by the HYDRA kernel is minimal (the most primitive "adequate" set) or that it is maximally desirable, we do believe the set provides primitives which are both necessary and adequate for the construction of a large and interesting class of operating environments. It is our view that the set of functions provided by HYDRA will enable the user of C.mmp to create his own operating environment without being confined to predetermined command and file systems, execution scenarios, resource allocation policies, etc.



If a kernel is to provide facilities for building an operating system and we wish to know what these facilities should be, then it is relevant to ask what an operating system *is* or *does*. Two views are commonly held: (1) an operating system defines an "abstract machine" by providing facilities, or resources, which are more convenient than those provided by the "bare" hardware; and (2) an operating system allocates (hardware) resources in such a way as to most effectively utilize them. Of course these views are, respectively, the bird's-eye and worm's eye views of what is a single entity with multiple goals. Nevertheless, the important observation for our purposes is the emphasis placed, in both views, on the central role of *resources*—both physical and abstract.

Current scoreboard

Red	Blue
1	3

Outline

- Hydra (cont.)
- The UNIX time-sharing operating system
- Mach: A New Kernel Foundation For UNIX Development

What HYDRA proposed

- Supporting multiple processors
- **Separation of mechanism and policy**
- Integration of the design with implementation methodology
- Rejection of strict hierarchical layering
- Protection
- Reliability

Poll close in 1:30

Who's policy?

- How many of the following terms belongs to "policies"?
  ① Least-recently used (LRU)
  ② First-in, first-out
  ③ Paging
  ④ Preemptive scheduling
  ⑤ Capability
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

Who's policy?

- How many of the following terms belongs to "policies"?
  - ① Least-recently used (LRU)
  - ② First-in, first-out
  - ③ Paging
  - ④ Preemptive scheduling
  - ⑤ Capability
  - A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

Who's policy?

- How many of the following terms belongs to "policies"?

  ① Least-recently used (LRU) — **Policy**

  ② First-in, first-out — **Policy**

  ③ Paging — **Mechanism**

  ④ Preemptive scheduling — **Mechanism**

  ⑤ Capability — **Mechanism**

  A. 0

  B. 1

  [✓ C. 2]

  D. 3

  E. 4

Impacts of HYDRA

- Flat system design to provide flexibility
- A unified abstraction of system resources (objects)
    - Object oriented programming
    - Protection mechanism — exists in many modern OSes with different implementations (will talk about this in Mach)

What the OS kernel should do?

The UNIX Time-Sharing System

Dennis M. Ritchie and Ken Thompson
Bell Laboratories

DENNIS RITCHIE & KEN THOMPSON

Inventors of UNIX.

acm

A.M. TURING AWARD 1983

Poll close in 1:30

Why they built "UNIX"

- How many of following statements is/are the motivations of building UNIX?
  ① Reducing the cost of building machines with powerful OSes
  ② Reducing the burden of maintaining the OS code
  ③ Reducing the size of the OS code
  ④ Supporting networks and multiprocessors
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

Why they built "UNIX"

- How many of following statements is/are the motivations of building UNIX?

  ① Reducing the cost of building machines with powerful OSes

  ② Reducing the burden of maintaining the OS code

  ③ Reducing the size of the OS code

  ④ Supporting networks and multiprocessors

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Why they built "UNIX"

- How many of following statements is/are the motivations of building UNIX?

  ① Reducing the cost of building machines with powerful OSes

  ② Reducing the burden of maintaining the OS code

  ③ Reducing the size of the OS code

  ④ Supporting networks and multiprocessors

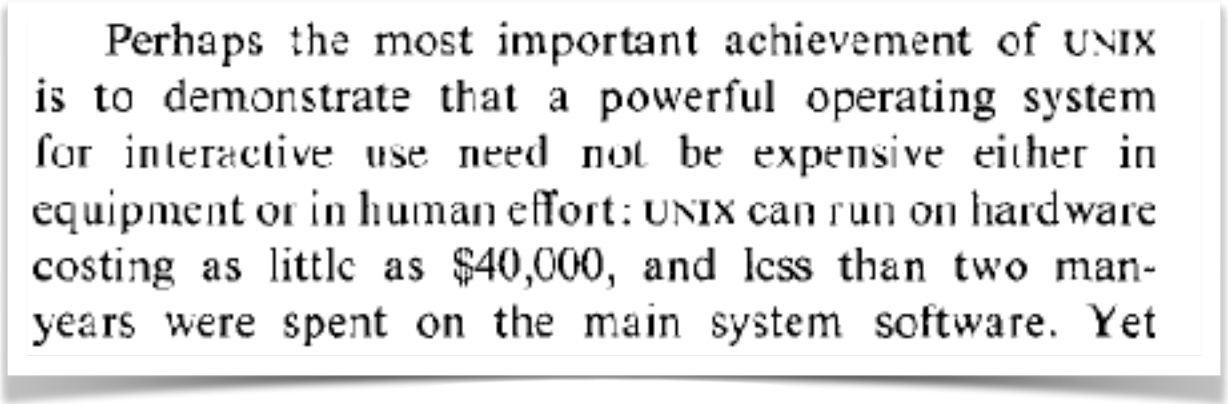
  A. 0

  B. 1

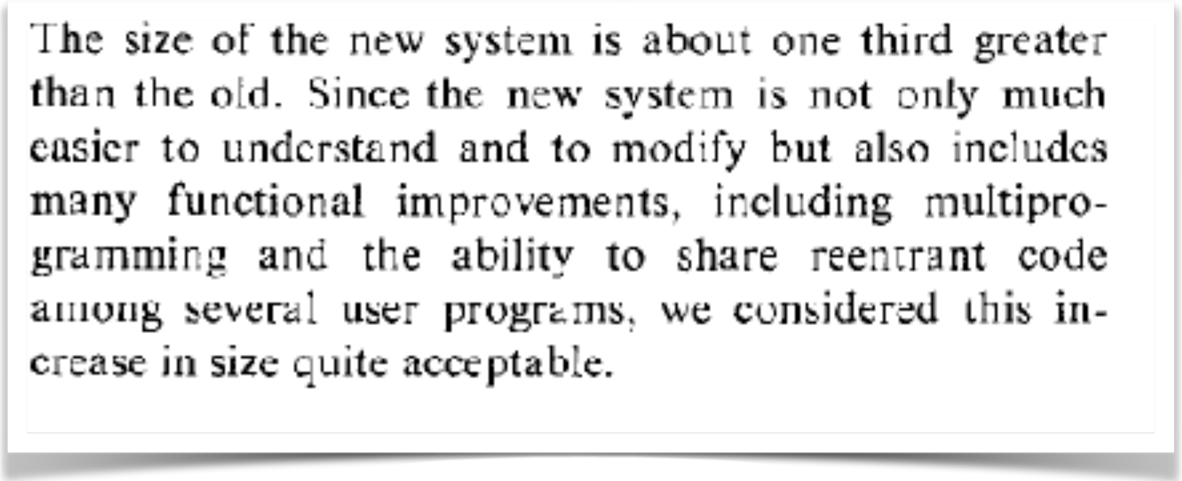
  [✓ C. 2]

  D. 3

  E. 4



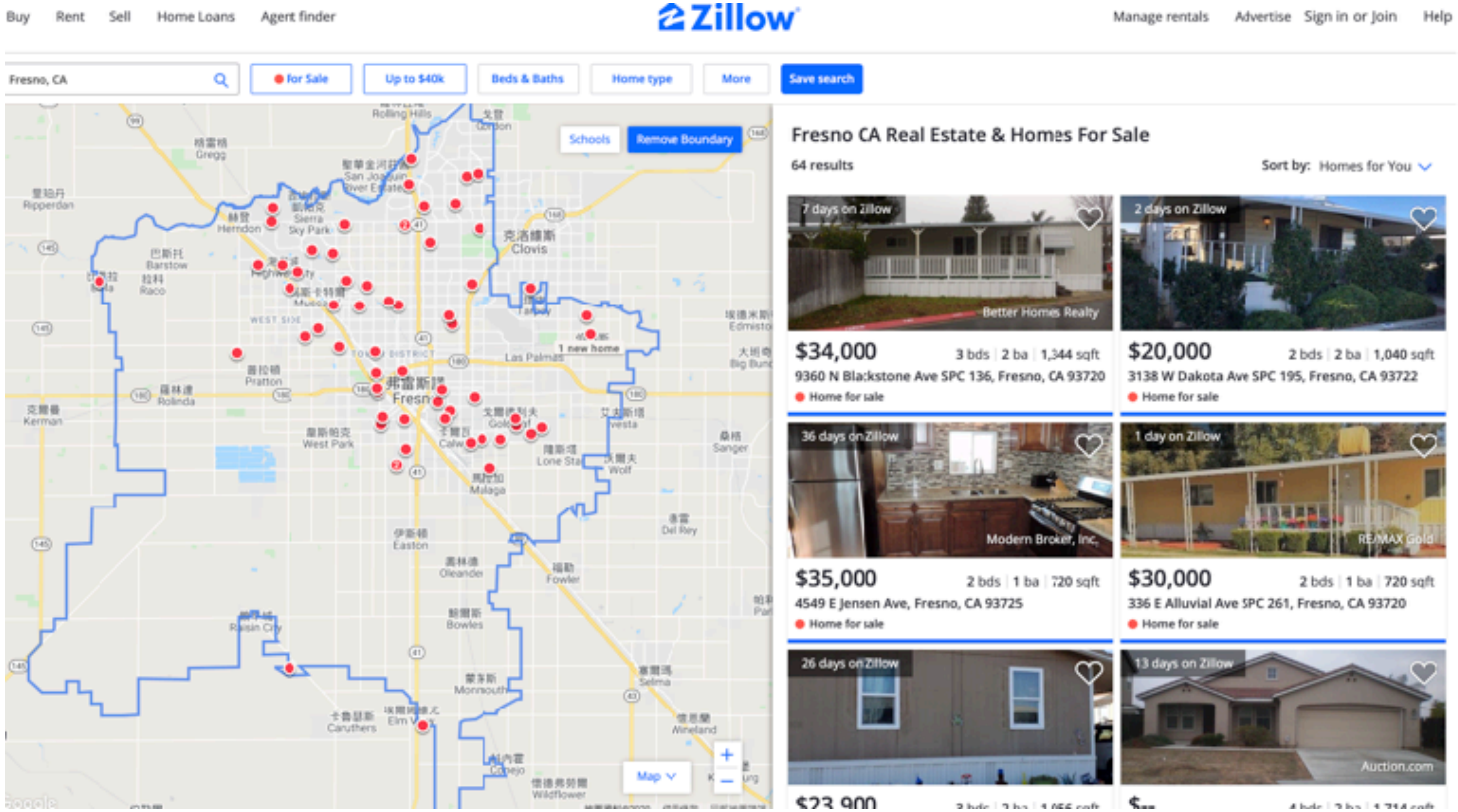
Perhaps the most important achievement of UNIX is to demonstrate that a powerful operating system for interactive use need not be expensive either in equipment or in human effort: UNIX can run on hardware costing as little as $40,000, and less than two man-years were spent on the main system software. Yet



The size of the new system is about one third greater than the old. Since the new system is not only much easier to understand and to modify but also includes many functional improvements, including multiprogramming and the ability to share reentrant code among several user programs, we considered this increase in size quite acceptable.

Why should we care about “UNIX”

- A powerful operating system on “inexpensive” hardware (still costs USD $40,000)
- An operating system promotes simplicity, elegance, and ease of use
- They made it



Fresno CA Real Estate & Homes For Sale

64 results

Sort by: Homes for You

$34,000 3 bds | 2 ba | 1,344 sqft
9360 N Blackstone Ave SPC 136, Fresno, CA 93720
● Home for sale

$20,000 2 bds | 2 ba | 1,040 sqft
3138 W Dakota Ave SPC 195, Fresno, CA 93722
● Home for sale

$35,000 2 bds | 1 ba | 720 sqft
4549 E Jensen Ave, Fresno, CA 93725
● Home for sale

$30,000 2 bds | 1 ba | 720 sqft
336 E Alluvial Ave SPC 261, Fresno, CA 93720
● Home for sale

$23,900 3 bds | 2 ba | 1,056 sqft

$ 4 bds | 2 ba | 1,714 sqft

What UNIX proposed

- Providing a file system
- File as the unifying abstraction in UNIX
- Remind what we mentioned before

Poll close in 1:30

The file abstraction

- How many of the following statements about UNIX is/are correct?
  ① The semantics of accessing a device and accessing a text file is the same
  ② For the file name `/alpha/beta/gamma`, alpha, beta, gamma are all files.
  ③ Altering the content of directory requires privileged operations
  ④ The programmer needs to treat random and sequential file accesses differently
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

The file abstraction

- How many of the following statements about UNIX is/are correct?
  ① The semantics of accessing a device and accessing a text file is the same
  ② For the file name `/alpha/beta/gamma`, alpha, beta, gamma are all files.
  ③ Altering the content of directory requires privileged operations
  ④ The programmer needs to treat random and sequential file accesses differently
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

The file abstraction

- How many of the following statements about UNIX is/are correct?

  ① The semantics of accessing a device and accessing a text file is the same

  ② For the file name `/alpha/beta/gamma`, alpha, beta, gamma are all files.

  ③ Altering the content of directory requires privileged operations

  ④ The programmer needs to treat random and sequential file accesses differently

  doesn't

  A. 0

  B. 1

  C. 2

  [✓ D. 3]

  E. 4

Poll close in 1:30

Protection

- Regarding the protection in the assigned UNIX paper, how many of the followings is/are correct?
  ① The same file may have different permissions for different user-id
  ② The owner of the file may not have the permission of writing a file
  ③ If the user does not have a permission to access a device, set-user-id will guarantee that the user will not be able to access that device
  ④ In the UNIX system described in this paper, if the file owner is "foo", then the user "bar" will have the same permission as another user (e.g. "xyz").
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

Protection

- Regarding the protection in the assigned UNIX paper, how many of the followings is/are correct?
  ① The same file may have different permissions for different user-id
  ② The owner of the file may not have the permission of writing a file
  ③ If the user does not have a permission to access a device, set-user-id will guarantee that the user will not be able to access that device
  ④ In the UNIX system described in this paper, if the file owner is "foo", then the user "bar" will have the same permission as another user (e.g. "xyz").
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Protection

- Regarding the protection in the assigned UNIX paper, how many of the followings is/are correct?

① The same file may have different permissions for different user-id

② The owner of the file may not have the permission of writing a file

③ If the user does not have a permission to access a device, set-user-id will ~~guarantee that the user will not be able to access that~~ device

allow the user to have the same permission as the creator of the

④ In the UNIX system described in this paper, if the file owner is "foo", then the user "bar" will have the same permission as another user (e.g. "xyz").

A. 0

The UNIX system at that time doesn't have "group" — everyone other than the owner is "others"

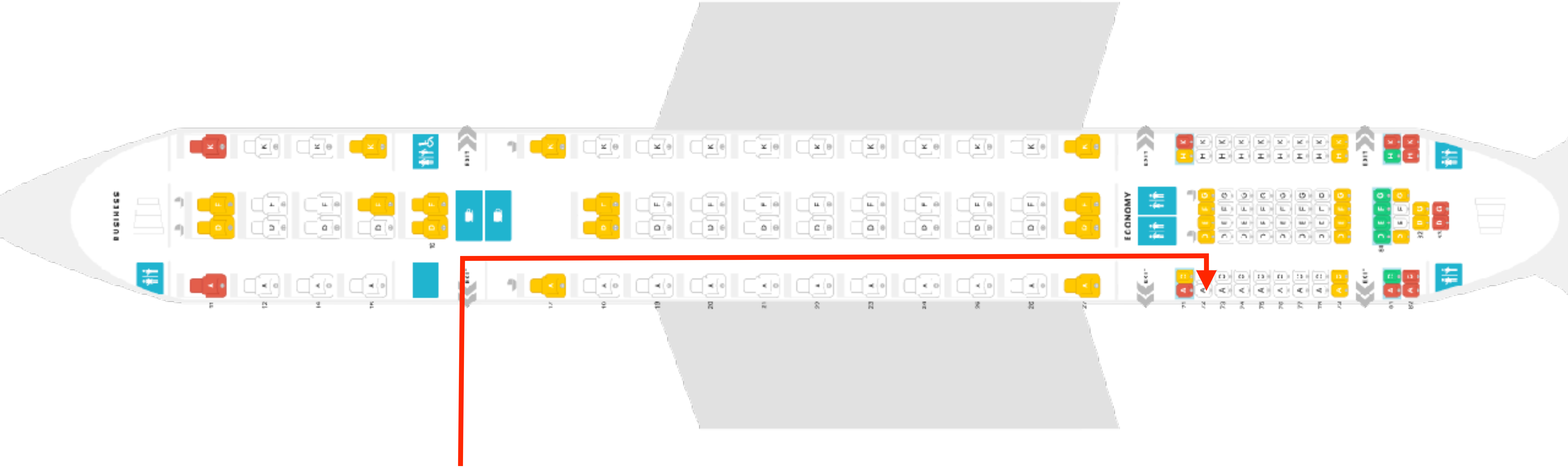
B. 1

C. 2

[✓ D. 3]

E. 4

Right amplification



Demo: setuid

- chmod u+s allows “others” to execute the program as the creator
- There exists a file “others” cannot read
- Another program can dump the content
- Without setuid, others still cannot read the content
- With setuid, others can read that!

Poll close in 1:30

What's in the kernel?

- How many of the following UNIX features/functions are implemented in the kernel?
  ① I/O device drivers
  ② File system
  ③ Shell
  ④ Virtual memory management
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

Poll close in 1:30

What’s in the kernel?

- How many of the following UNIX features/functions are implemented in the kernel?

  ① I/O device drivers

  ② File system

  ③ Shell

  ④ Virtual memory management

  A. 0

  B. 1

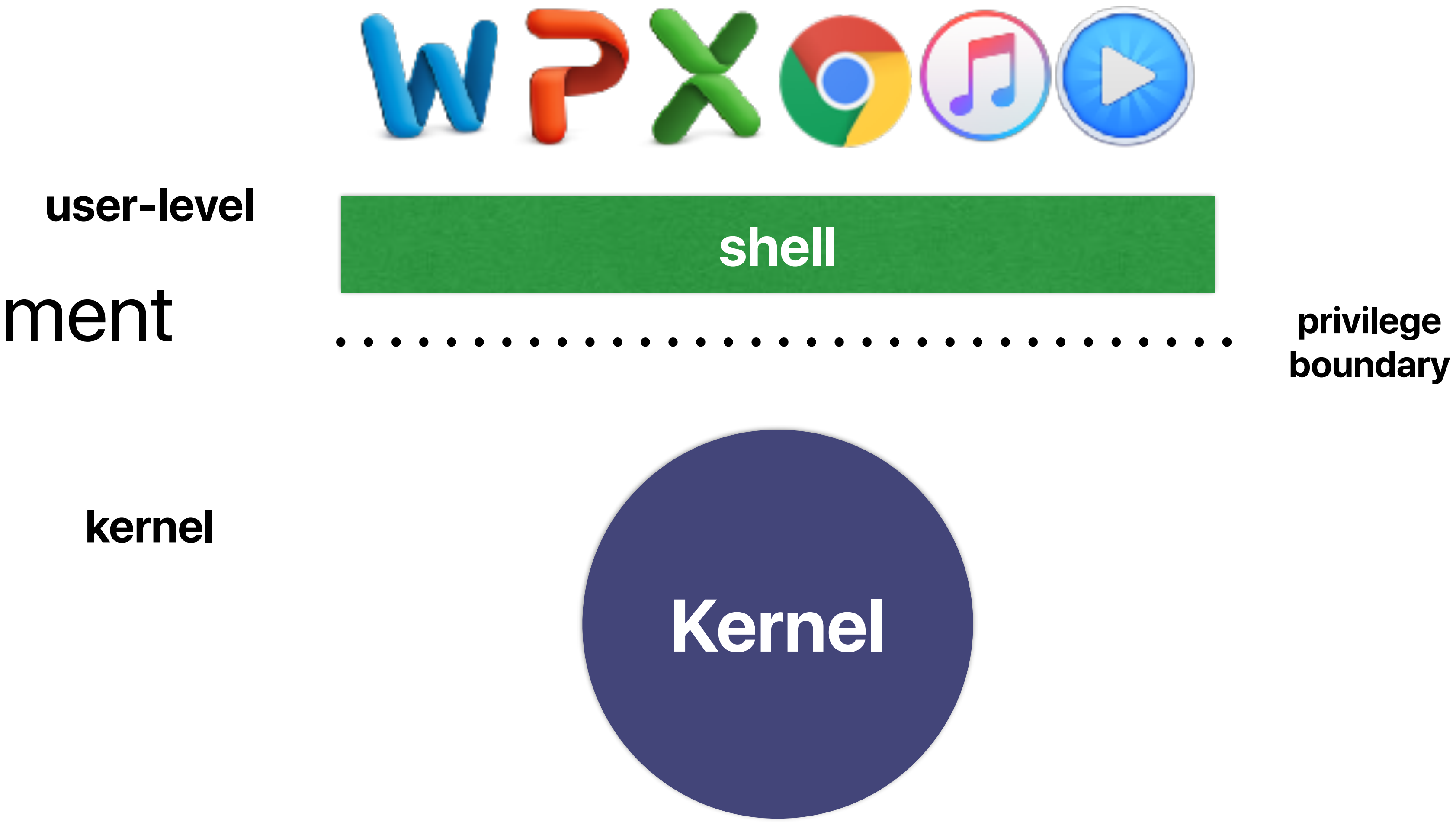
  C. 2

  D. 3

  E. 4

What's in the kernel?

- How many of the following UNIX features/functions are implemented in the kernel?
  ① I/O device drivers
  ② File system
  ③ Shell
  ④ Virtual memory management
  A. 0
  B. 1
  C. 2
  [✓ D. 3]
  E. 4



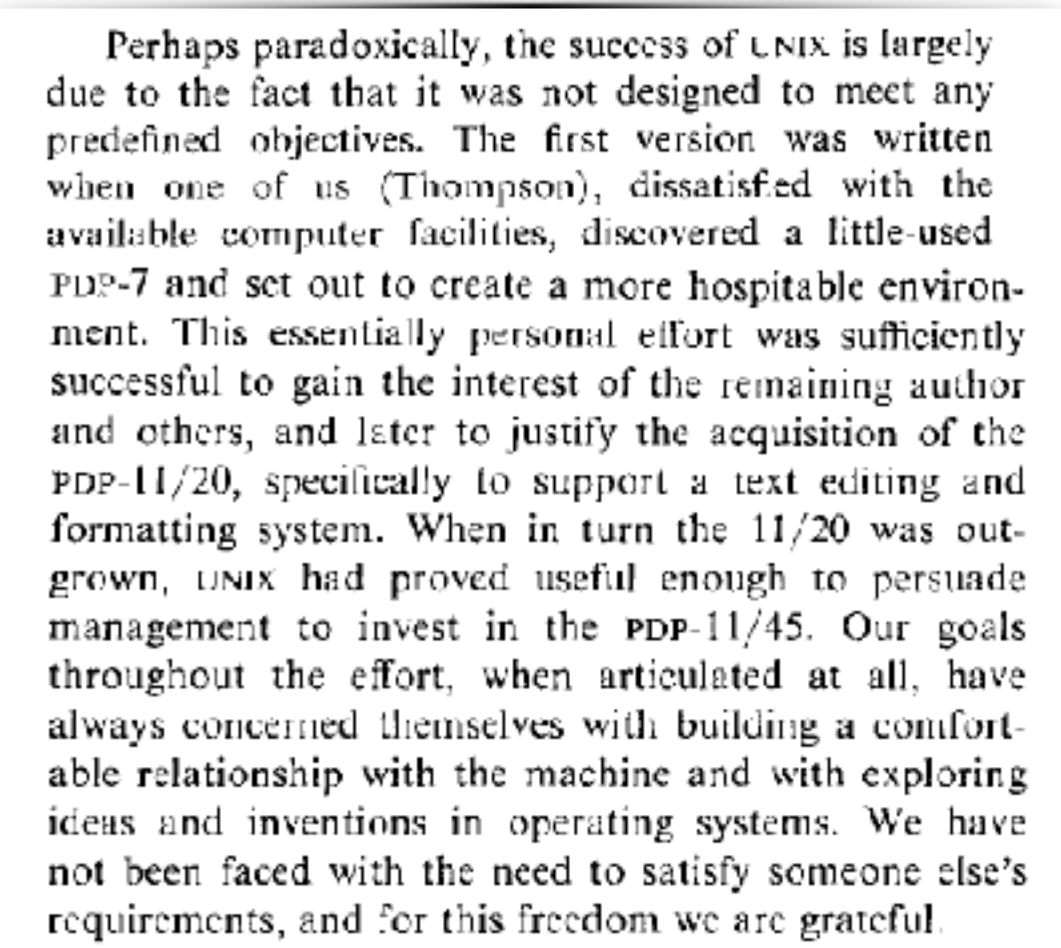
Shell

- A user program provides an interactive UI
- Interprets user command into OS functions
- Basic semantics:

  command argument_1 argument_2 …
- Advanced semantics
  - Redirection
    - >
    - <
  - Pipe
    - |
  - Multitasking
    - &

The impact of UNIX

- Clean abstraction
- File system — will discuss in detail after midterm
- Portable OS
  - Written in high-level C programming language
  - The unshakable position of C programming language
- We are still using it!



Perhaps paradoxically, the success of UNIX is largely due to the fact that it was not designed to meet any predefined objectives. The first version was written when one of us (Thompson), dissatisfied with the available computer facilities, discovered a little-used PDP-7 and set out to create a more hospitable environment. This essentially personal effort was sufficiently successful to gain the interest of the remaining author and others, and later to justify the acquisition of the PDP-11/20, specifically to support a text editing and formatting system. When in turn the 11/20 was outgrown, UNIX had proved useful enough to persuade management to invest in the PDP-11/45. Our goals throughout the effort, when articulated at all, have always concerned themselves with building a comfortable relationship with the machine and with exploring ideas and inventions in operating systems. We have not been faced with the need to satisfy someone else's requirements, and for this freedom we are grateful.

Announcement

- Reading quizzes due next Tuesday
  - Welcome new friends! — will drop a total of 6 reading quizzes for the quarter
  - Attendance count as 4 reading quizzes
- Change of office hour next week — W 9a-11a (since Monday is MLK day)
- Project groups in 2
  - Will release the project by the end of the next week

Computer
Science &
Engineering

202

つづく