



CPU

CPU  
CPU

(Z\$ #

CPU

CPU

I/O

CPU

I/O

user mode

kernel mode

trap

return-from-trap

trap table

I/O

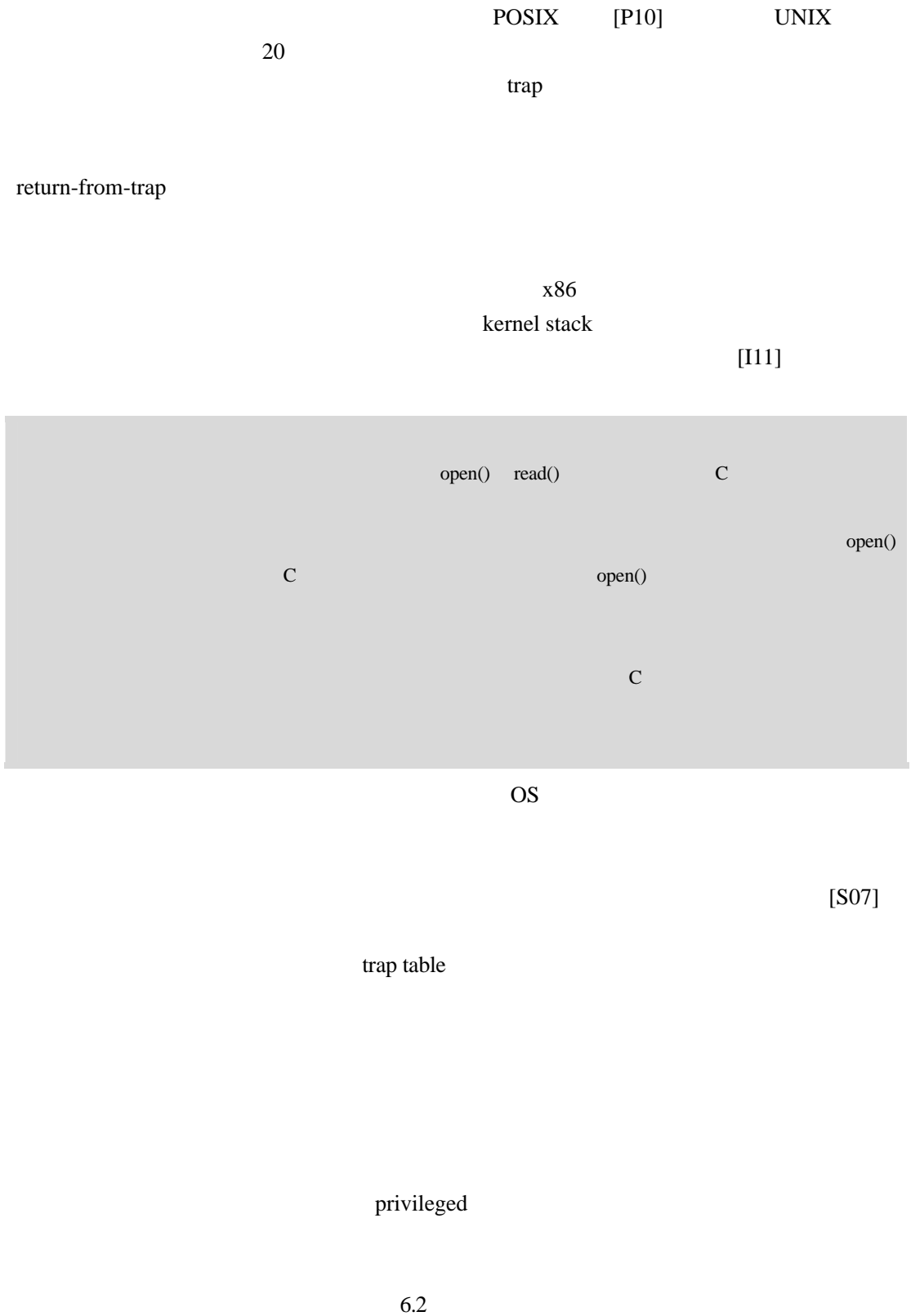
I/O

user mode

I/O

kernel mode

I/O



(Z\$

@		
@		
argv /		
	main	
		main
		main exit()

LDE

CPU

CPU

main()

exit()

OS

OS

(Z%

\$

CPU

CPU

CPU

CPU

regain control

Xerox Alto [A79]

cooperative

Macintosh [M11]

CPU

CPU

CPU

yield

0

CPU

trap

OS

CPU

CPU



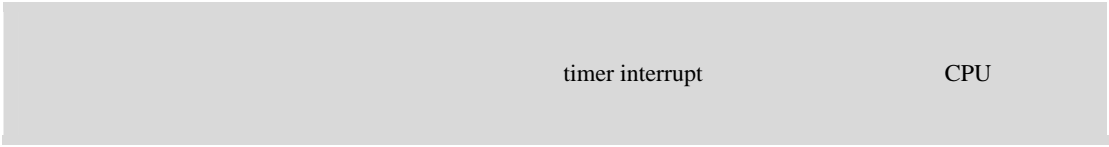
CPU

timer interrupt

[M+63]

interrupt handler

CPU



timer interrupt

CPU

scheduler

OS

context switch

6.3	A	
	A B	switch()
	A B	B
switch context		B
A	B	
(Z%		
@		
	x ms CPU	
@		
		A
	A A	
switch()		
A B A B		
B B		
	B B	
	B	
		B

/

A

B

OS

A

B

6.1 xv6

x86

xv6

context

old

new

```

1  # void swtch(struct context **old, struct context *new);
2  #
3  # Save current register context in old
4  # and then load register context from new.
5  .globl swtch

```

```

6   swtch:
7       # Save old registers
8       movl 4(%esp), %eax # put old ptr into eax
9       popl 0(%eax)      # save the old IP
10      movl %esp, 4(%eax) # and stack
11      movl %ebx, 8(%eax) # and other registers
12      movl %ecx, 12(%eax)
13      movl %edx, 16(%eax)
14      movl %esi, 20(%eax)
15      movl %edi, 24(%eax)
16      movl %ebp, 28(%eax)
17
18      # Load new registers
19      movl 4(%esp), %eax # put new ptr into eax
20      movl 28(%eax), %ebp # restore other registers
21      movl 24(%eax), %edi
22      movl 20(%eax), %esi
23      movl 16(%eax), %edx
24      movl 12(%eax), %ecx
25      movl 8(%eax), %ebx
26      movl 4(%eax), %esp # stack is switched here
27      pushl 0(%eax)      # return addr put in place
28      ret                # finally return into new ctxt

```

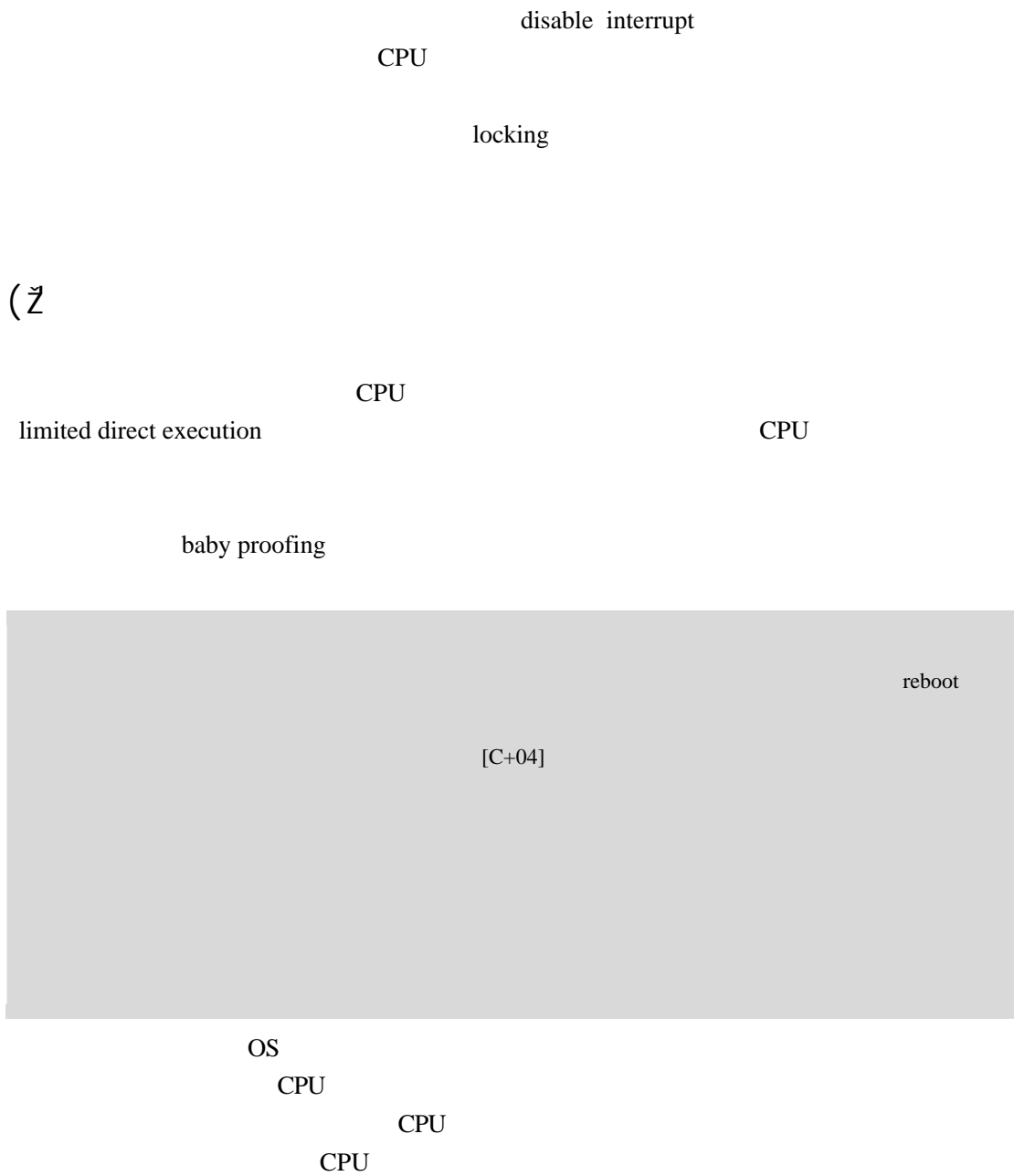
## 6.1 xv6

( $\mathbb{Z}$ )

2

Imbench [MS96]				1996	200-MHz P6
CPU	Linux 1.3.37	4σs	6σs[MS96]		
		2 GHz	3 GHz		
		CPU	Ousterhout		
					[O90]





x

Lisa

Mac

[C+04] Microreboot ~~IA~~ A Technique for Cheap Recovery

George Candea, Shinichi Kawamoto, Yuichi Fujiki, Greg Friedman, Armando Fox OSDI ~~04~~, San Francisco, CA, December 2004

[I11] Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3A and 3B: System Programming Guide

Intel Corporation, January 2011

[K+61] One-Level Storage System

T. Kilburn, D.B.G. Edwards, M.J. Lanigan, F.H. Sumner IRE Transactions on Electronic Computers, April 1962  
Atlas

[L78]

[L78] The Manchester Mark I and Atlas: A Historical Perspective

S. H. Lavington

Communications of the ACM, 21:1, January 1978

Atlas

[M+63] A Time-Sharing Debugging System for a Small Computer

J. McCarthy, S. Boilen, E. Fredkin, J. C. R. Licklider AFIPS ~~63~~ (Spring), May, 1963, New York, USA

17

[MS96] Imbench: Portable tools for performance analysis Larry McVoy and Carl Staelin

USENIX Annual Technical Conference, January 1996

Imbench

[M11] macOS 9

January 2011

[O90] Why Aren't Operating Systems Getting Faster as Fast as Hardware?

J. Ousterhout

USENIX Summer Conference, June 1990

[P10] The Single UNIX Specification, Version 3 The Open Group, May 2010

[S07] The Geometry of Innocent Flesh on the Bone: Return-into-libc without Function Calls (on the x86) Hovav Shacham

CCS '07, October 2007

