第 2 讲：OS Architecture & Structure

第四节：Micro kernel – Mach & L4

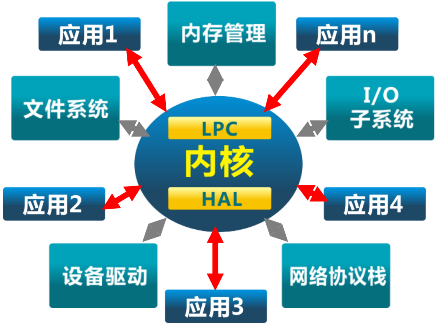
陈渝

清华大学计算机系

yuchen@tsinghua.edu.cn

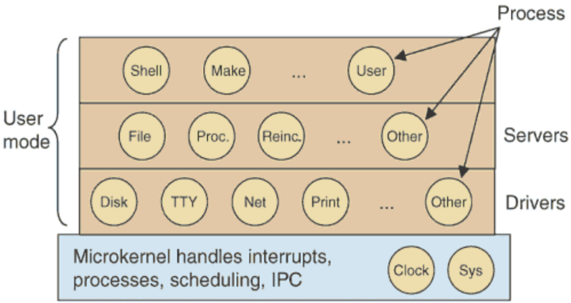
2020 年 2 月 23 日

What is a Microkernel?



- Kernel with minimal features
- Moves as much from the kernel into user space
  - Address spaces
  - Interprocess communication (IPC)
  - Scheduling

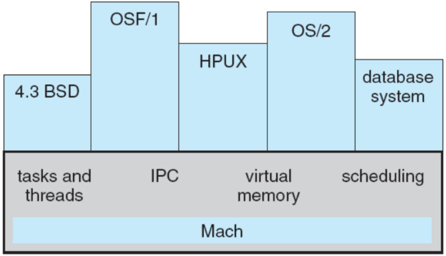
What is a Microkernel?



- Benefits
  - Flexibility
  - Safety
  - Modularity
- Detriments
  - Address spaces
  - Interprocess communication (IPC)
  - Scheduling

Microkernel– Mach

Mach Developed at CMU Led by Rick Rashid, Founded Microsoft Research Initial release: 1985



4.3 BSD

OSF/1

HPUX

OS/2

database system

tasks and threads

IPC

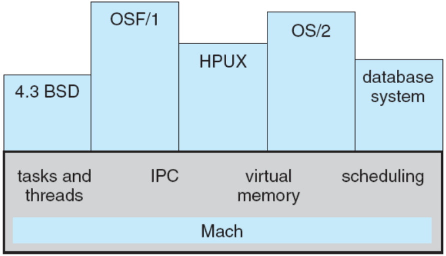
virtual memory

scheduling

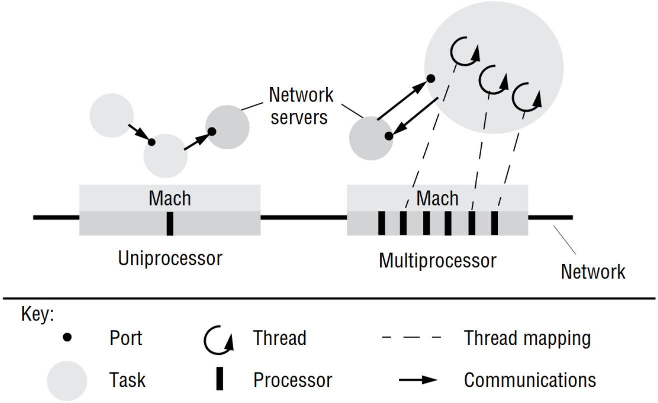
Mach

- First generation microkernel – Mach (led by Rick Rashid)
- Task and thread management
  - Task (process) unit of allocation
  - Thread, unit of execution
  - CPU scheduling policies exposed to apps
- Interprocess communication (IPC)
  - Between threads via ports
  - Secured by capabilities
- Memory object management
  - virtual memory
  - memory object
  - hierarchical pagers

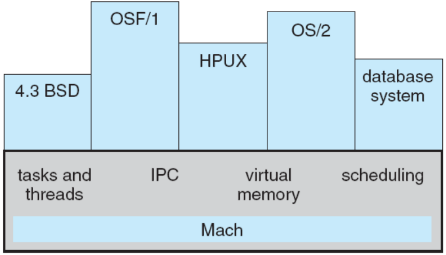
Microkernel– Mach



- First generation microkernel – Mach

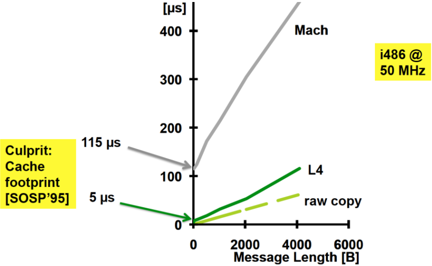


Microkernel– Mach



- First generation microkernel – Mach
- Performance
  - the use of IPC for almost all tasks turned out to have serious performance impact.
  - system calls take 5-6X as long as UNIX
  - given a syscall that does nothing, a full round-trip under BSD would require about 40μs, whereas on a user-space Mach system it would take just under 500μs.
  - benchmarks on 1997 hardware showed that Mach 3.0-based UNIX single-server implementations were about 50% slower than native UNIX.

Microkernel– L4

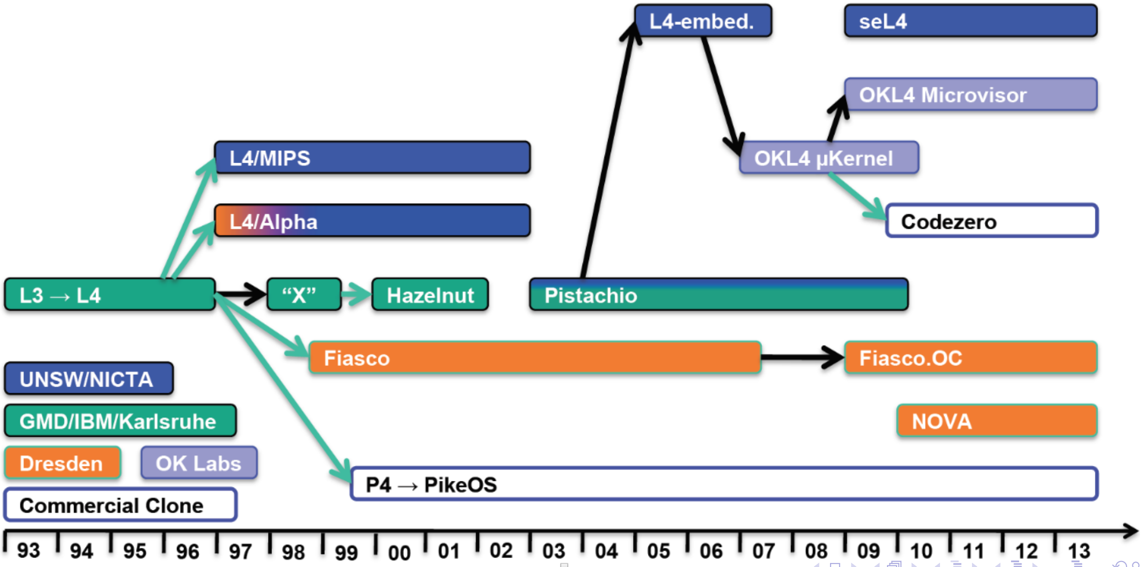


Culprit: Cache footprint [SOSP'95]

i486 @ 50 MHz

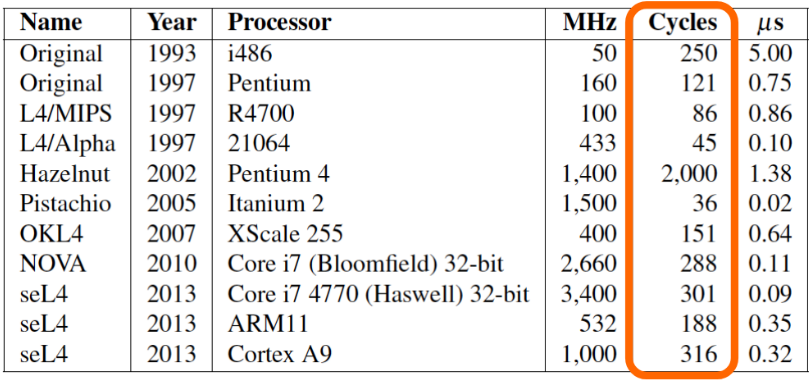
- Second generation microkernel – L4 by Jochen Liedtke (GMD)
- Performance
  - synchronous IPCs –> async IPCs (like epoll in Linux)
  - smaller, Mach 3(330 KB) –> L4 (12KB)
  - IPC security checks moved to user process
  - IPC is hardware dependent

L4 family


L4-embed.
seL4
OKL4 Microvisor
L4/MIPS
OKL4 µKernel
L4/Alpha
Codezero
L3 → L4
“X”
Hazelnut
Pistachio
Fiasco
Fiasco.OC
UNSW/NICTA
GMD/IBM/Karlsruhe
NOVA
Dresden
OK Labs
P4 → PikeOS
Commercial Clone
93
94
95
96
97
98
99
00
01
02
03
04
05
06
07
08
09
10
11
12
13

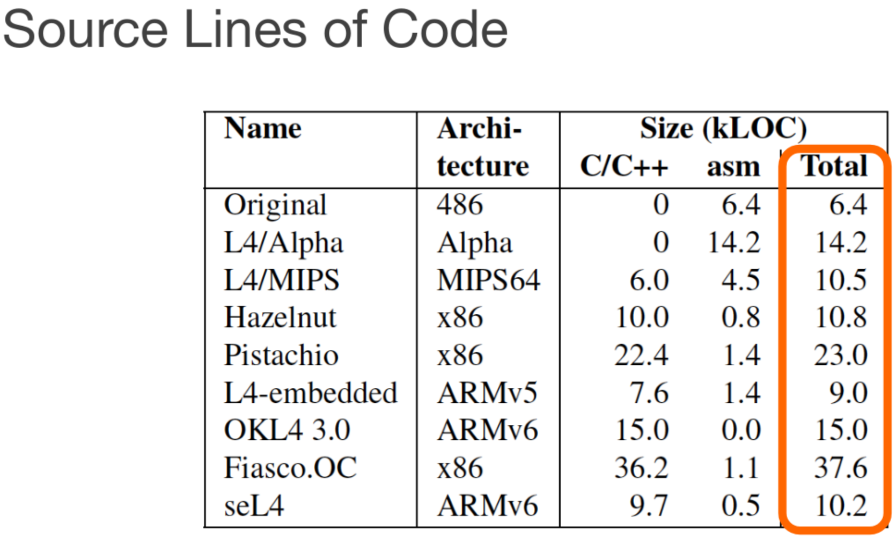
L4 family

One-way IPC cost over years



Name	Year	Processor	MHz	Cycles	μs
Original	1993	i486	50	250	5.00
Original	1997	Pentium	160	121	0.75
L4/MIPS	1997	R4700	100	86	0.86
L4/Alpha	1997	21064	433	45	0.10
Hazelnut	2002	Pentium 4	1,400	2,000	1.38
Pistachio	2005	Itanium 2	1,500	36	0.02
OKL4	2007	XScale 255	400	151	0.64
NOVA	2010	Core i7 (Bloomfield) 32-bit	2,660	288	0.11
seL4	2013	Core i7 4770 (Haswell) 32-bit	3,400	301	0.09
seL4	2013	ARM11	532	188	0.35
seL4	2013	Cortex A9	1,000	316	0.32

L4 family



Source Lines of Code

Name	Architecture	Size (kLOC)		
		C/C++	asm	Total
Original	486	0	6.4	6.4
L4/Alpha	Alpha	0	14.2	14.2
L4/MIPS	MIPS64	6.0	4.5	10.5
Hazelnut	x86	10.0	0.8	10.8
Pistachio	x86	22.4	1.4	23.0
L4-embedded	ARMv5	7.6	1.4	9.0
OKL4 3.0	ARMv6	15.0	0.0	15.0
Fiasco.OC	x86	36.2	1.1	37.6
seL4	ARMv6	9.7	0.5	10.2