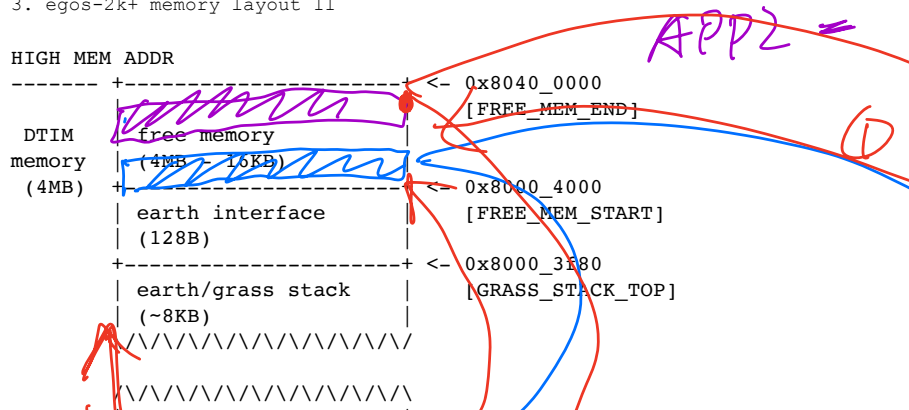


3. egos-2k+ memory layout II

HIGH MEM ADDR		
DTIM memory (4MB)	free memory (4MB - 16KB)	0x8040_0000 [FREE_MEM_END]
	earth interface (128B)	0x8000_4000 [FREE_MEM_START]
	earth/grass stack (~8KB)	0x8000_3f80 [GRASS_STACK_TOP]
	grass interface	0x8000_2000 [APPS_STACK_TOP]
	app stack (6KB)	0x8000_0800
	system call args (1KB)	0x8000_0400 [SYSCALL_ARG]
	app args (1KB)	0x8000_0000 [APPS_ARG]
...		...
	read-only fs w/ dirs & user app elfs (2MB)	0x20c0_0000
disk image (4MB)	grass elf and sys app elfs (1MB)	0x20a0_0000
	(1MB)	0x2090_0000
boot ROM (4MB)	earth code	0x2080_0000
		0x2040_0000

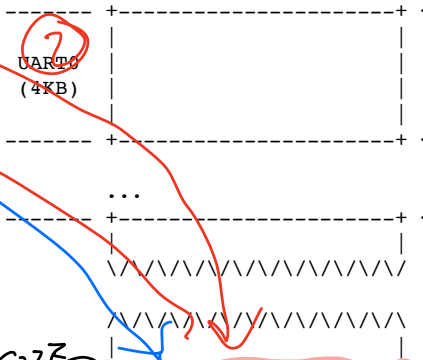
APP1  
APP2



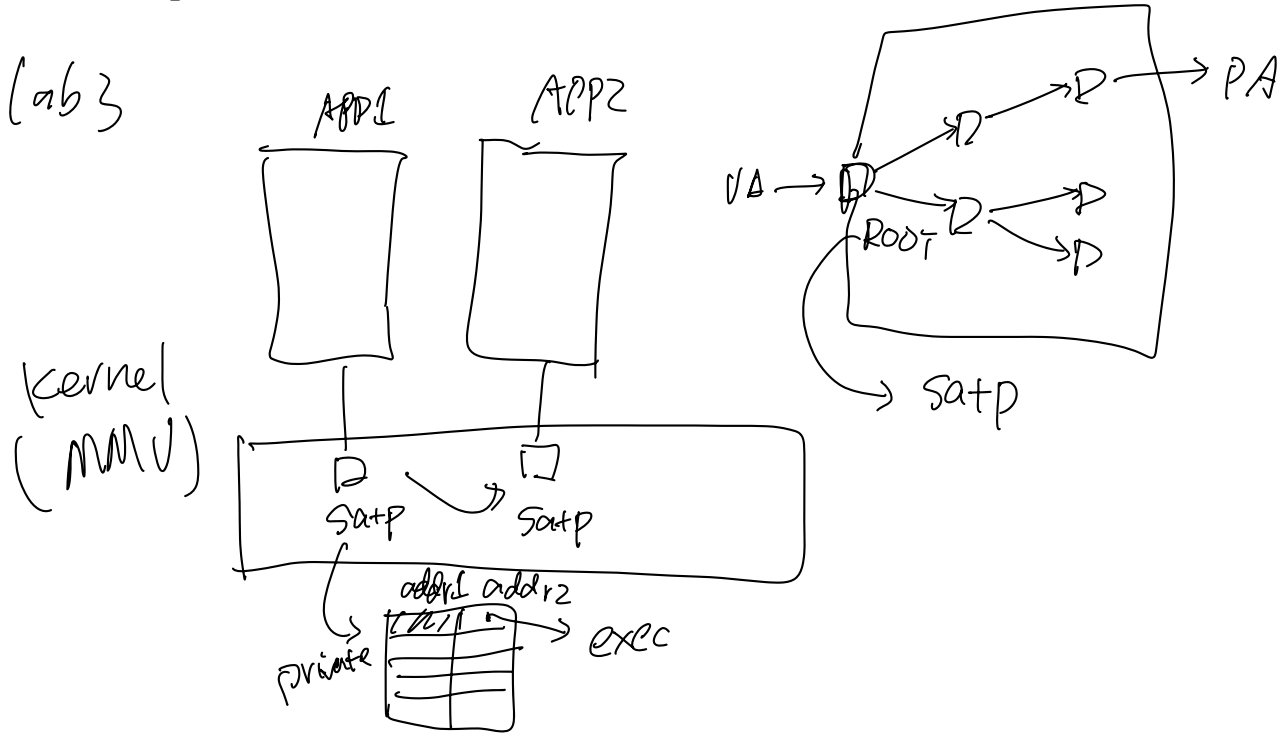
APP SIZE  
8KB  
APP SIZE  
4KB + 2

- ① Saving APP1's Mem
- ② copy APP2 to APPS\_ENTRY

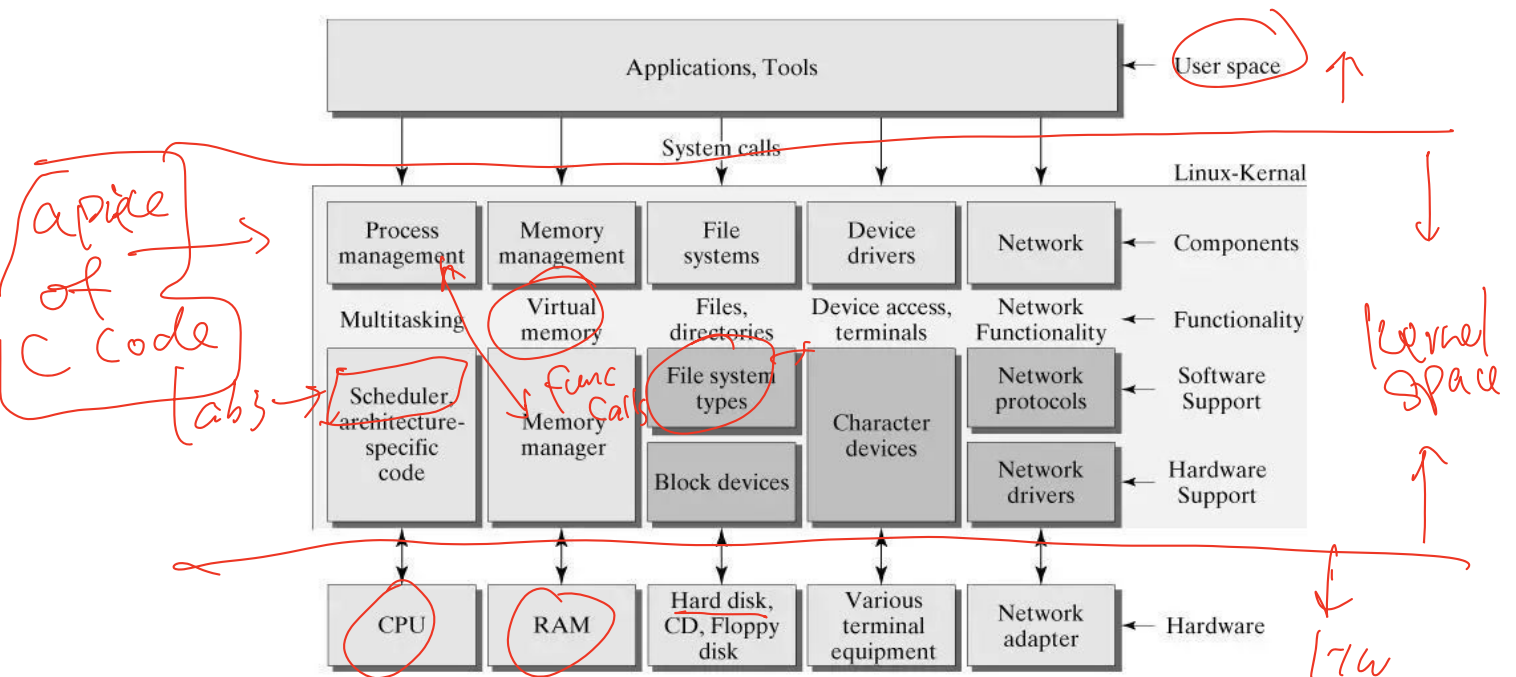
...			0x1001_4000
	UART0 (4KB)		0x1001_3000
...			...
			0x0a00_0000 [ITIM_END]
			0x0820_4000 [APPS_ENTRY+APPS_SIZE]
	ITIM (32MB)	app code+data (16KB)	0x0820_0000 [APPS_ENTRY]
		grass code+data (1 MB)	0x0810_0000 [GRASS_ENTRY]
		earth data (1 MB)	0x0800_0000 [ITIM_START]
...			...
			0x0201_0000
		mtime (8B)	0x0200_bff8
CLINT (64KB)			0x0200_4000
		mtimecp (8B)	0x0200_0000
		msip (4B)	
LOW MEM ADDR			



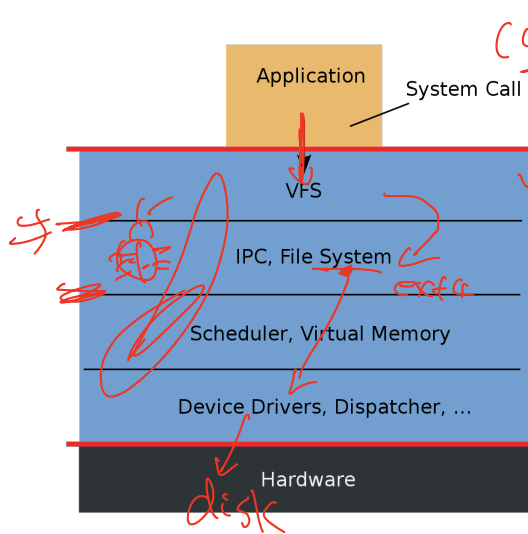
1. OSI: the first 3 steps
  2. OS organization ←
  3. egos desgin
  4. egos-2k+ implementation
- 



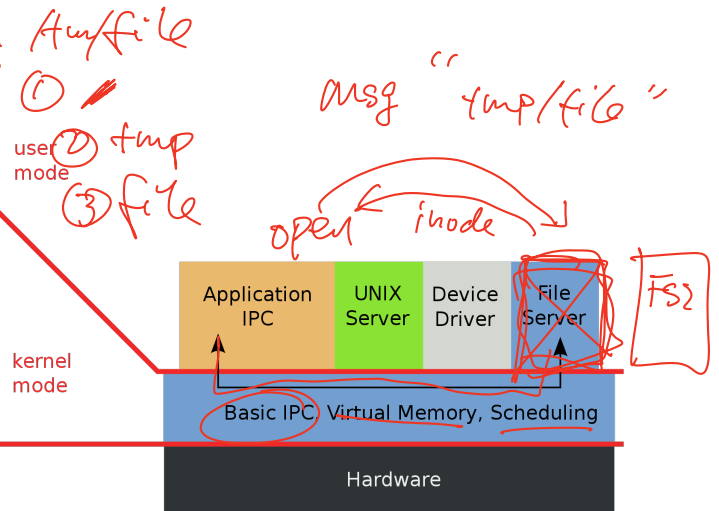
## Linux kernel architecture



## Monolithic Kernel based Operating System

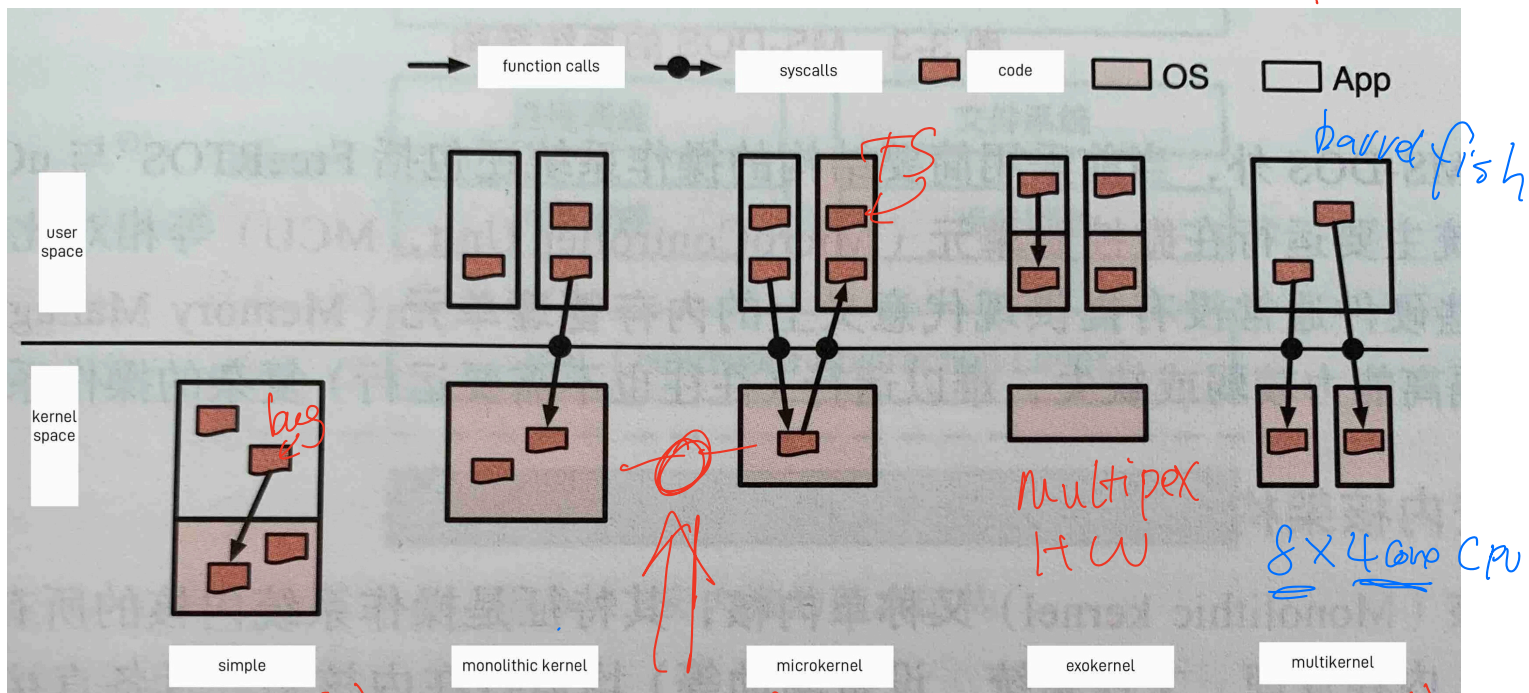


## Microkernel based Operating System



UNIX v6 10K  
Linux 0.01 8K

Linux 5.7	15K	50K	200K	500K	28M
Windows XP		30K	1M	1M	45M
Windows 8		50K	10M	10M	60M



(DOS)

(1970+)

(1995)

(2009)

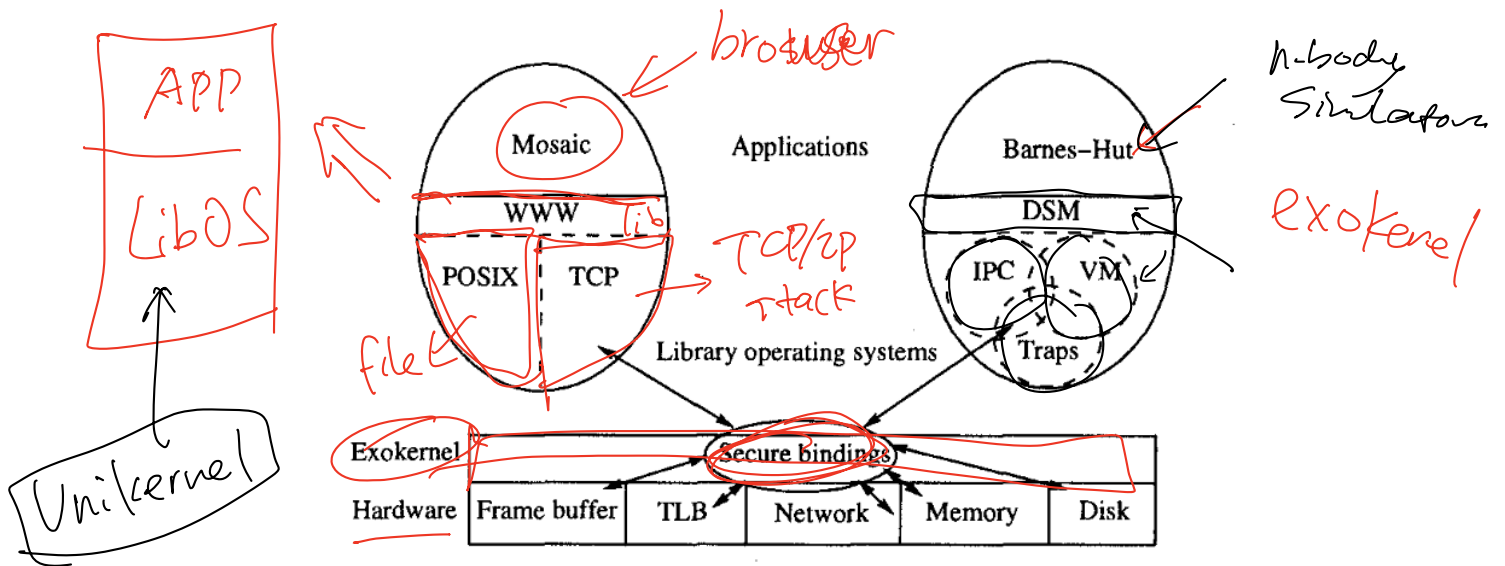


Figure 1: An example exokernel-based system consisting of a thin exokernel veneer that exports resources to library operating systems through secure bindings. Each library operating system implements its own system objects and policies. Applications link against standard libraries (e.g., WWW, POSIX, and TCP libraries for Web applications) or against specialized libraries (e.g., a distributed shared memory library for parallel applications).

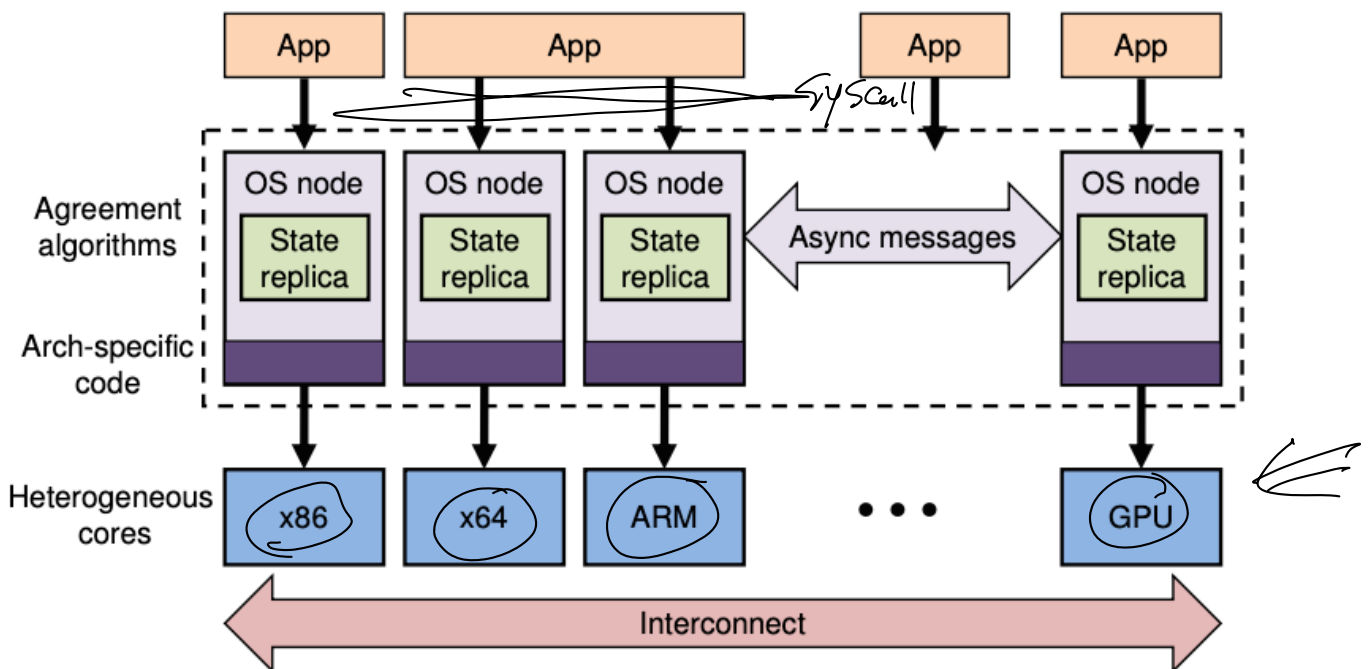


Figure 1: The multikernel model.

**Monolithic:**

a single piece of code serving all requests; high in coupling

**Microkernel:**

kernel is mostly responsible for IPCs; services are running in user-level; low in coupling; IPC can be the bottleneck

**Exokernel:**

kernel only handles multiplexing resources (securely); extra performance due to having hardware primitives

**Multikernel:**

target heterogeneous hardware and many-core machines  
replace shared-memory model with shared-nothing model (msg passing)  
treat OS as a distributed system

宏内核

微内核

外核

多内核

1960s

**Monolithic**

1980s

**Microkernel**

1990s

**Exokernel**

2010s

**Multikernel**

UNIX

VMS

Mach

Minix

Disco/SplashOS

Dune

Barrelfish

PopCorn

BSD

WinNT

L3

L4

Fiasco

MirageOS

IX

Vxworks

Arrakis

Linux

Mac OS X

coreOS

Pistachio

VMware

OSv

Corey

LibOS

RHEL

Ubuntu

SUSE

Cumulus

QNX

OKL4

图 3-9

the evolution of OS organization and architecture

