Operating Systems

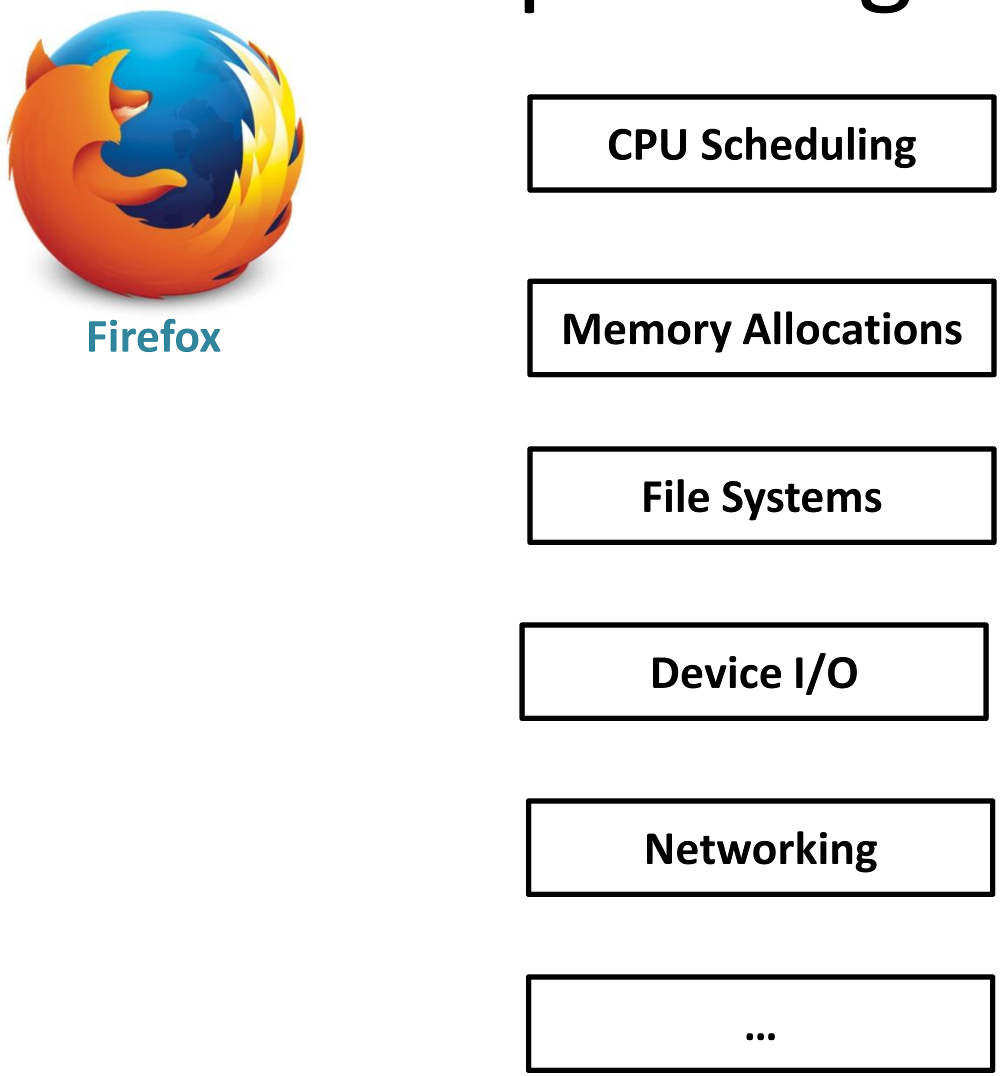
CS 241

September 4, 2013

Operating Systems

Firefox

Operating Systems



Firefox

CPU Scheduling

Memory Allocations

File Systems

Device I/O

Networking

…

Operating Systems

Firefox

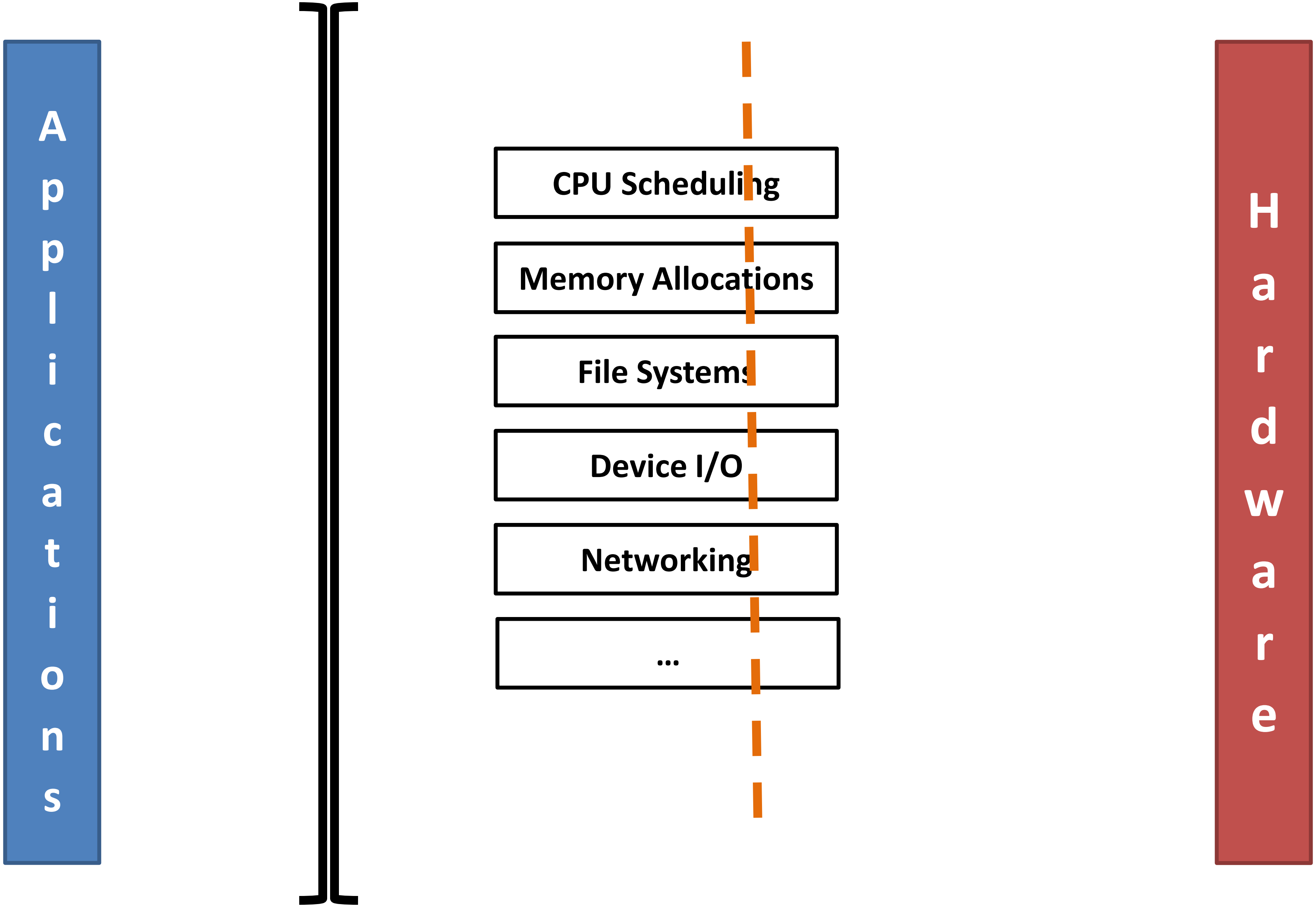
CPU Scheduling	x86	x64	ARM	…
Memory Allocations	RAM			
File Systems	SDCard	NTFS	ext4	…
Device I/O	Display	nVidia	ATI	Intel
	Mouse	USB	PS/2	…
	Keyboard	USB	PS/2	…
Networking	Wi-Fi	Ethernet	Mobile	…
…	IPC	Printing	Security	…

Operating Systems

Firefox

CPU Scheduling	x86	x64	ARM	…	
Memory Allocations	RAM				
File Systems	SDCard	NTFS	ext4	…	
Device I/O	Display	nVidia	ATI	Intel	
	Mouse	USB	PS/2	…	
	Keyboard	USB	PS/2	…	
Networking	Wi-Fi	Ethernet	Mobile	…	
…	IPC	Printing	Security	…	

Operating Systems



Applications

CPU Scheduling

Memory Allocations

File Systems

Device I/O

Networking

…

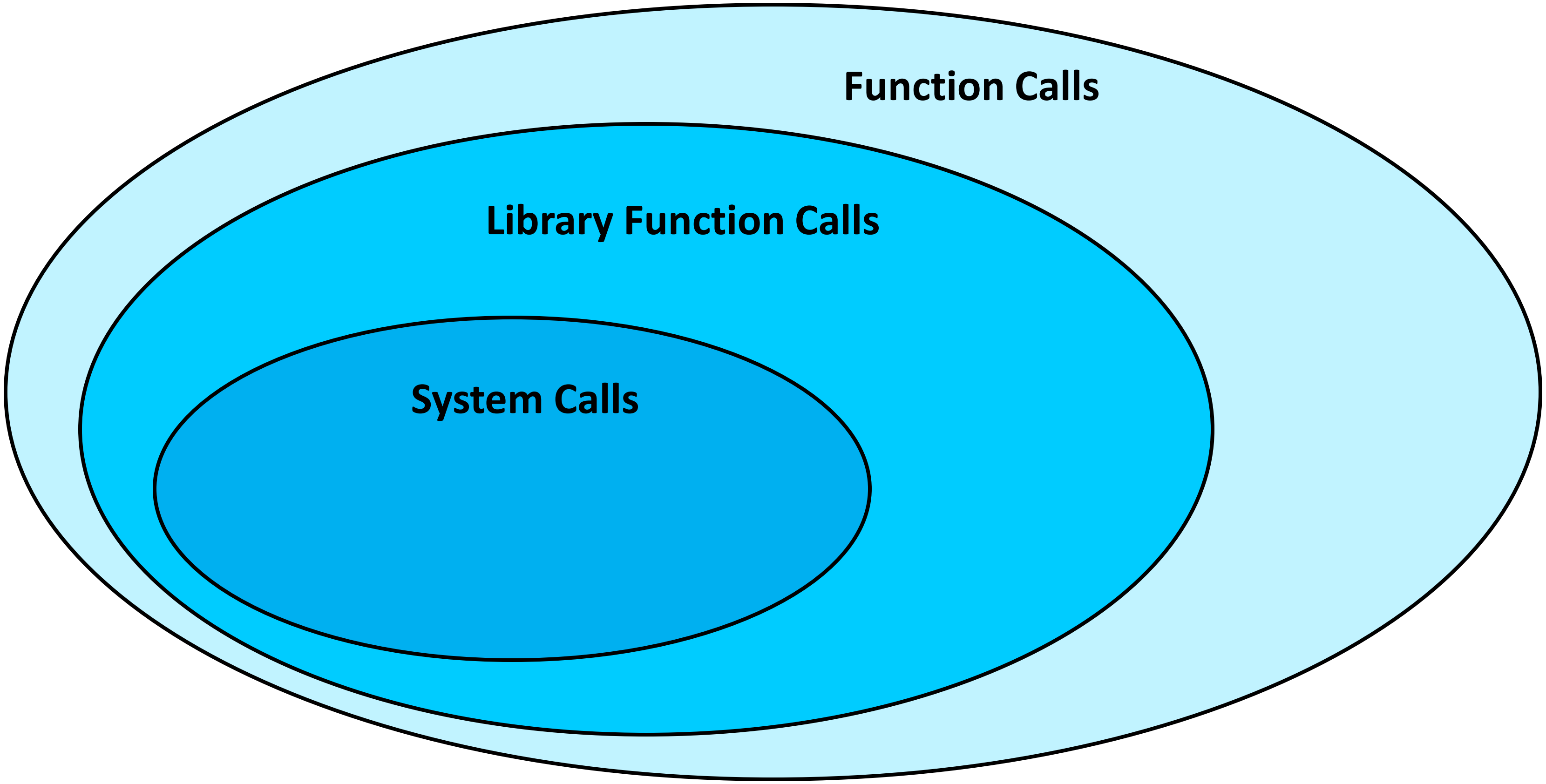
Hardware

How do we interact with the OS?

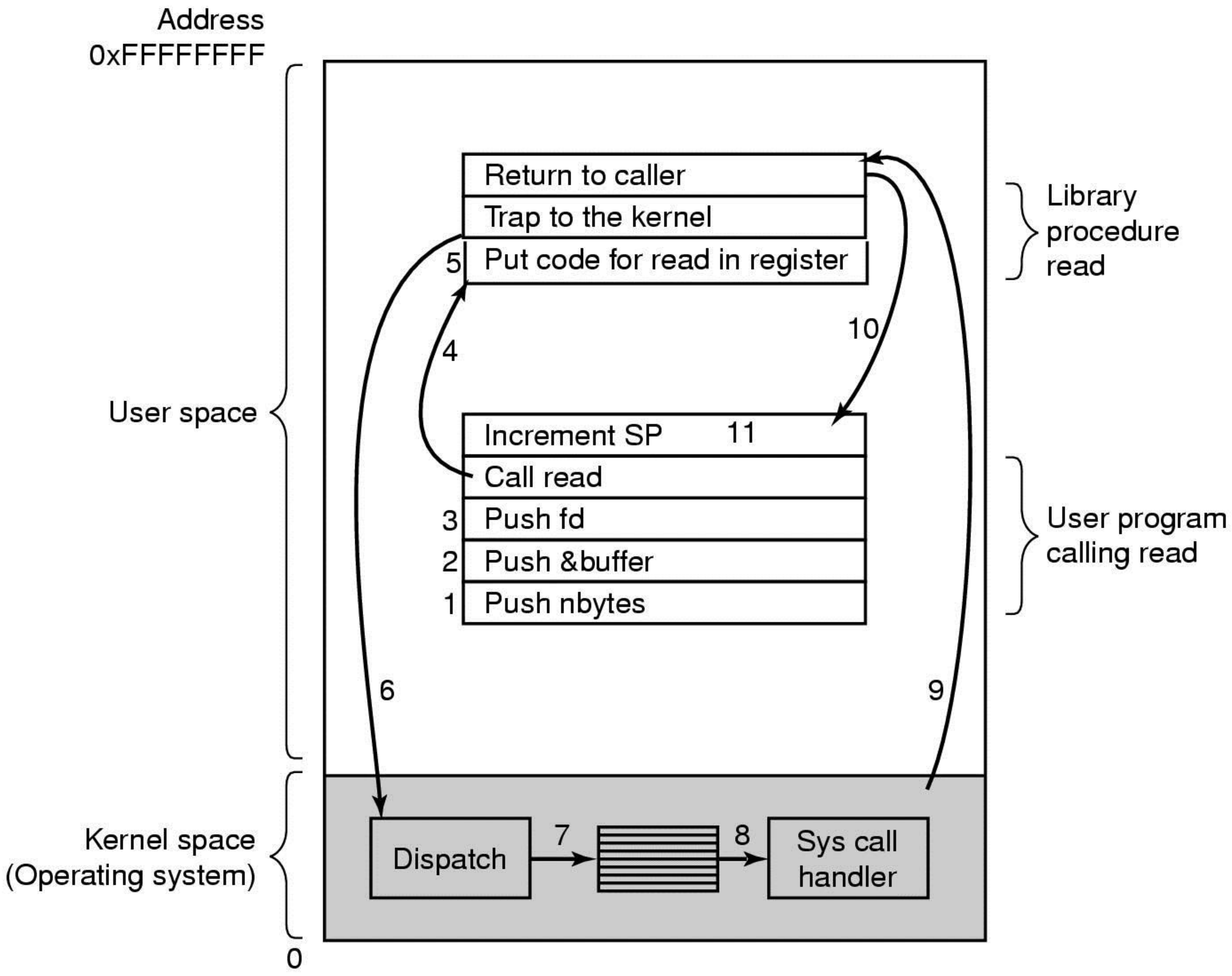
- **System Calls** allows an application to access a OS-managed resource
  - OS-managed resources can only be accessed in a privileged “kernel” level

Programmer's View

- Three types of function calls:
  - (Application) Function call
  - **Library** function call
  - **System** calls / syscall()


Function Calls
Library Function Calls
System Calls

```
count = read (fd, buffer, nbytes);
```



System Calls

- Almost all system calls fit within five broad categories:
  - Process Control
  - File Management
  - Device Management
  - Information Management
  - Communication

System Calls

- Process Control
  - `fork()` : Creates a child process
  - `exec()` : Execute a new process image
  - `kill()` : Terminate/signal a process
  - `wait()` : Wait for a process to complete
  - `sbrk()` : Increase process’ heap size
  - …

System Calls

- File Management
  - `open()` : Opens a file
  - `close()` : Closes a file
  - `read()` : Reads from a file
  - `write()` : Writes to a file
  - `lseek()` : Seek within a file
  - …

System Calls

- Device Management
  - **`mkdir()`** : Makes a directory
  - **`rmdir()`** : Removes an empty directory
  - **`link()`** : Creates a link to a file/directory
  - **`unlink()`** : Removes the link
  - **`mount()`** : Mount a device/file system
  - **`unmount()`** : Removes the mount
  - …

System Calls

- Information Management
  - **`stat()`**: Get status of a file/directory
  - **`times()`**: Process running times
  - **`getrusage()`**: Resource usage
  - **`clock_gettime()`**: Get system time
  - **`clock_getres()`**: Clock resolution
  - …

System Calls

- Communication
  - **`pipe()`**: Communicate b/t two processes
  - **`shmget()`**: Share memory b/t processes
  - **`mmap()`**: Maps virtual memory
  - **`socket()`**: Network socket
  - **`connect()`**: Connect to a remote server
  - **`accept()`**: Accept remote connection
  - **`send()`**: Send network messages
  - …

C Library Calls

```
#include <stdio.h>

char *s = "Hello";
printf("%s\n", s);
```