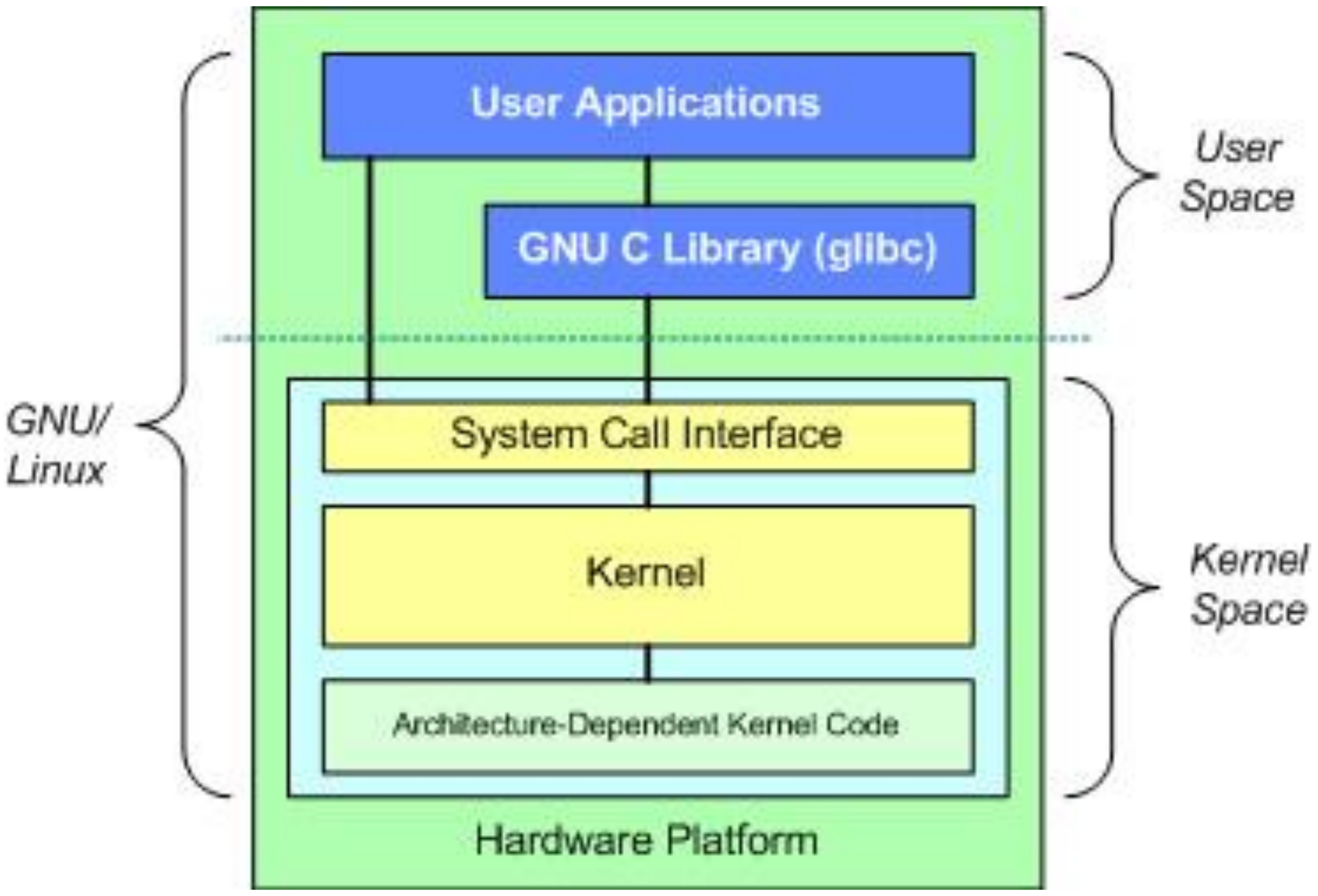
System Calls

Using Tanenbaum's Modern Operating Systems (3rd edition)

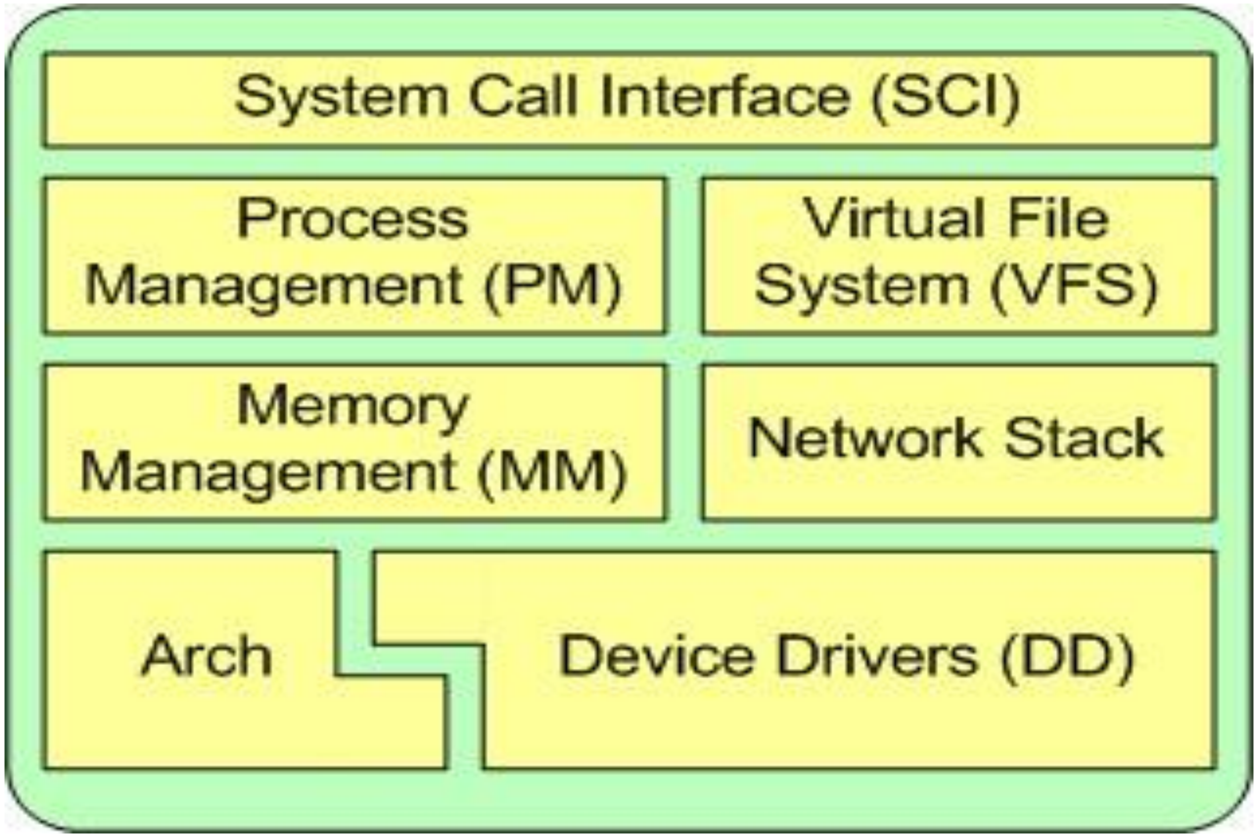
System Call

- What is a system call?
  - API to the OS
  - Requests a service from an operating system's kernel
- Two CPU modes :
  - User mode:
    - Limits the address space of the program.
    - Prevents the application from directly using devices.
  - Kernel mode:
    - All is allowed.
- POSIX defines about 100 system calls:
  - open, read, write, close, wait, execve, fork, exit, kill

Linux OS Architecture



Linux OS Architecture (II)



glibc

- The C standard library provides macros, type definitions, and functions:
  - Basic manipulation:
    - String handling.
    - Mathematical computations.
  - System calls:
    - I/O processing.
    - Memory allocation.
    - Other OS services.

glibc (II)

- By calling a system-call:
  - Sets the parameters and TRAP instruction, which:
    - Switches into kernel mode – transfers control to OS
    - Jumps to a single fixed location

- "*fork*" and "*execve*" are *glibc* functions that in turn call the "fork" and "execve" system-calls.

System Calls (I)

- Process control:
  - load
  - execute
  - create process
  - terminate process
  - get/set process attributes
  - wait for time, wait event, signal event
  - allocate, free memory

System Calls (II)

- File management:
  - create file, delete file
  - open, close
  - read, write, reposition
  - get/set file attributes
- Device management:
  - attach or detach devices
  - read, write, reposition
  - get/set device attributes

System Calls (III)

- Information:
  - get/set time or date
  - get/set system data
  - get/set process, file, or device attributes
- Communications:
  - create, delete communications connection
  - send, receive messages
  - transfer status information
  - attach or detach remote devices

System Call Flow

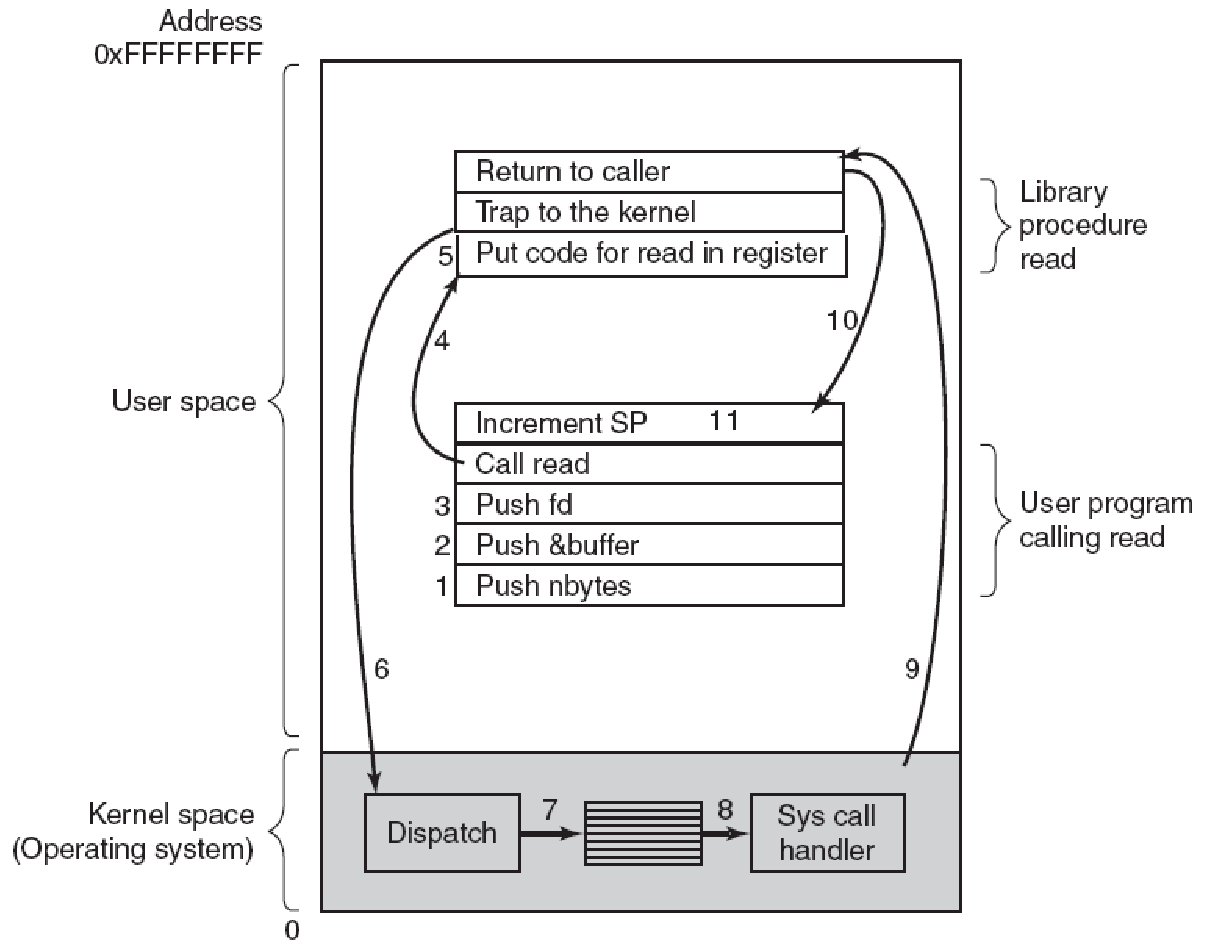
- C program:

  ```
  …
  count = read(fd,buffer,nbyts);
  …
  ```

  - counts is the number of read bytes.
  - if failed count=-1 and *errno* is set.

- The library function *read* is called
- The system call number is put into a register
- A TRAP instruction is executed to switch to kernel mode
- The kernel starts at a fixed address
- The kernel executes the system call handler that is a function pointer assigned to the specific system call number
- The handler is executed and control is returned to the user-space library at the instruction after the TRAP

System Call (11 steps)



Blocking System Call

Example:

- The user calls *read*() from the keyboard but nothing has been typed yet.
- What happens?
  - The system call will run the kernel code for blocking the process until a keyboard interrupt arrives
  - In the meantime, another process is taken from the Ready Queue to be executed

POSIX System Calls (I)

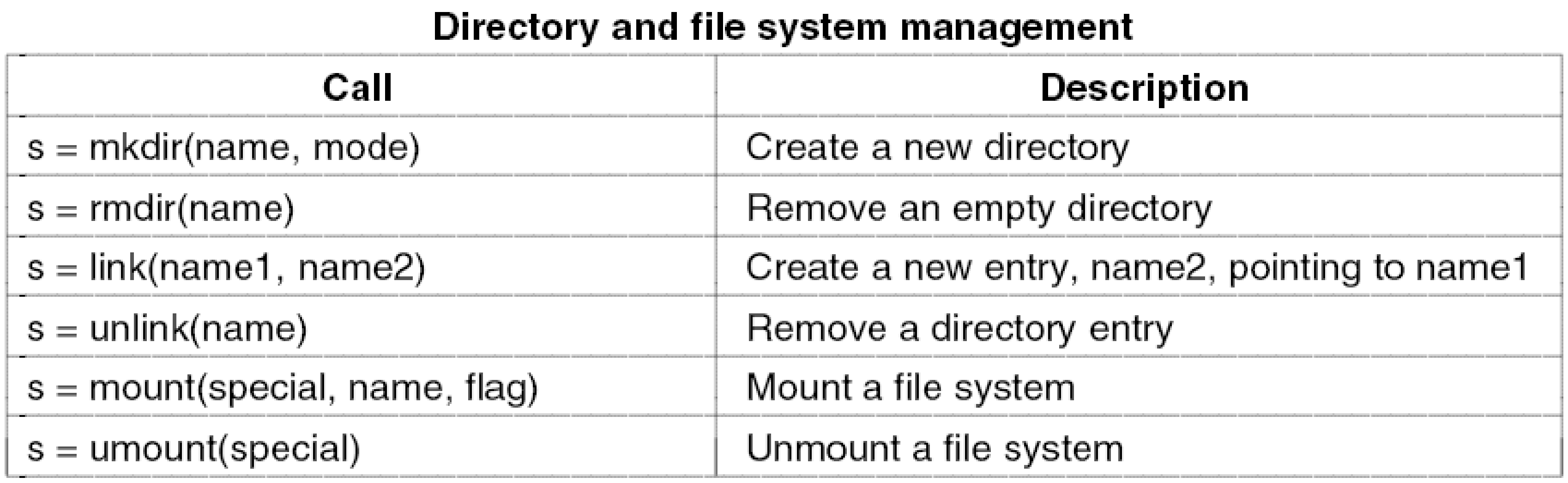
Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management

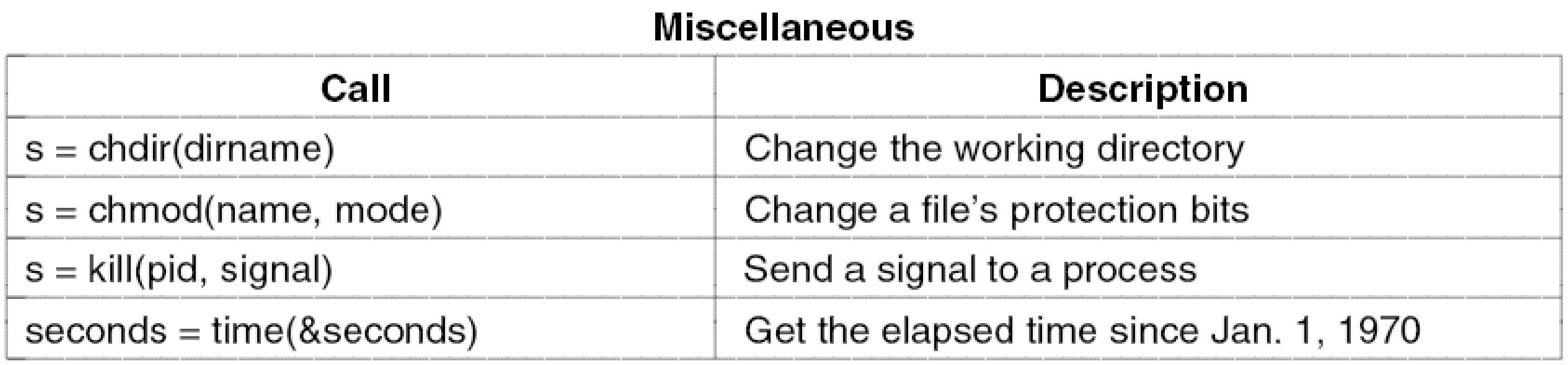
Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

POSIX System Calls (II)



Directory and file system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system



Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Homework + Interview Questions

- What is “system call”?
- Explain in detail how system call is executed
- What is TRAP?