Jacob Khalili & John Pluchino

Operating Systems PS 1

Problem 1 – What is kernel mode / what is user mode?

1. putchar()- always will call write, because it writes to stdout. – wrong if buffer filled
2. calloc()- will always trigger a system call; however brk or sbrk are used for small requests but mmap is used if the memory request is large. – only if u need more space, malloc/calloc keeps track of open area in the heap that you already have it
3. strchr() - will not call any system calls as it does not require any supervisor permissions as it accepts an a char \* that should exist in the scope of the program.
4. getting the current time of day - will always utilize a system call that is named gettimeofday, because the program would not have access to this information without attaining it from the kernel.

Problem 2 -- Errors and error messages

(a) close(-1)

i. Because -1, will never be a valid file descriptor as -1 is designated for error.

ii. EBADF

iii. Bad file descriptor

(b) We call write to output 4096 bytes to a file on disk, but the disk is completely full

i.You don’t have space to store more data so writing will not be possible.

ii. ENOSPC

iii. No space left on device

(c) int fd = open("/bogus",O\_RDONLY); /\* Note -- "/bogus" does not exist \*/

i. This is an error because open cannot perform its usual function as the file that it is attempting to open and return a file descriptor for does not exist.

ii. ENOENT

iii. No such file or directory

(d) We call read on a file which was previously successfully opened for reading. The file resides on a USB stick. After we opened it, the user pulled the USB stick out.

i. Because the process is unable to write to the external device, because it is not plugged in.

ii. EIO

iii.Input/output error