Derek Lee, Shine Li, Steven Lee Computer Operating Systems Fall 2020

Professor Hakner PSET #1

Problem 1

a) Requires a system call. The input is put into a buffer. Once the buffer is full, read() is called, and functions can copy characters out of the buffer until it becomes empty again.
b) Depends. Malloc may require a system call depending on how much memory is remaining in the heap. If the available memory is too small or too fragmented, it will make a system call to get more memory, using brk().[1]
c) Does not require a system call. Assuming a process has already been assigned to a program, and that the requisite conditions for running the program have been met (assignment of virtual memory, access to the CPU, etc.), then the contents of the program (including the functions) have already been loaded into the RAM allocated for the process. Calling the function involves merely directing the CPU to the start of the set of instructions associated with the function.
d) Requires a system call. We know this because UNIX uses a system call to get the time. It uses the system call gettimeofday() .
e) Does not require a system call. This is because there exist mathematical algorithms to compute the digits of π which can be implemented inside of a C function, and as established in (Question 1c), the act of calling a function does not require a system call. These algorithms involve purely arithmetical operations (which can be done using the CPU), and nothing else (no I/O etc.).

[1] https://stackoverflow.com/questions/6326474/so-malloc-doesnt-invoke-any-syscall

Problem 2

a) n = -1. It is writing to an invalid file. (return errono has EBADF)

b) Returns a file descriptor to a newly created file, truncated to length 0, with universal read/write permissions.

c) It prints out the number of bytes read until it reaches the EOF. Once it reaches the EOF, it prints out 0. In this case, it will print out 33300000000...

d) n = -1. errno = 30. The file is read only.