Christopher Brant
EEL 5737
Due 9/4/19

HW1 Part B Responses

1) ls – lists items in a directory
   touch – creates new files or updates the timestamp of an existing file
   echo – displays/prints a line of text or a string to stdout
   cat – concatenates a file and default prints to stdout
   cp – copies files or directories
   rm – removes files or directories

2) ls – fstat() is the system call chosen for ls, and this system call retrieves information on a file, which is important for listing items or item info in a directory. The system call fstat() has two arguments, the first is an integer file descriptor, and the second is a pointer to a buffer in which to place the statistics structure. The return value for this system call is an integer, 0 on success and -1 on error.

   touch – open() is the system call chosen for touch, and this system call opens a file for viewing or editing and can create a new file if the given filename does not exist. This is important for changing file info or creating new files the way the touch command does. The system call open() has two arguments, a pathname argument for what to open, and a flags argument for any necessary flags. It returns a new integer file descriptor on success and a -1 on error.

   echo – write() is the system call chosen for echo, which writes data to a file/location in memory, whether that is a device file, a normal file, or stdout (as everything in linux is treated as a file). This is important for writing output to the screen as it writes output to a file, and for echo that file would be stdout. The system call write() has three arguments, an integer file descriptor, a pointer to a buffer containing data to write, and a count which is the number of bytes to write. It returns an integer of the number of bytes written upon success, 0 if nothing was written, and -1 on error.

   cat – read() is the system call chosen for cat, which reads data from a given memory location/file. This is important for the cat command as it is necessary for reading data from a file such that the cat command can concatenate the contents of that memory location/file to output to the screen or be piped to another file. The system call read() has 3 arguments just like write, which are again, an integer file descriptor, a pointer to a buffer to read the data into, and a count of the number of bytes to read. It returns an integer of the number of bytes read upon success, and a -1 on error.

   cp – fadvise64() is the system call chosen for cp, which advises the kernel about the intended pattern of accessing data for a specific file in the future. This is important such that the OS can optimize the actions for handling that file in the future, so for the cp command, it would let the OS know that this file will be accessed with a certain pattern

since it is either creating or writing over an existing file. The system call fadvise64() has four arguments, an integer file descriptor, an offset into that file/memory location, a length of region of data that it is advising upon, and then an integer designating the specific pattern that it is advising the OS is intended for that file. It returns an integer of a 0 on success and a -1 on error.

rm – unlinkat() is the system call chosen for rm, which deletes a specific filename from the filesystem, which is important for the rm command as when you delete/remove a file, that filename should obviously be removed from the filesystem. The unlinkat() system call has three arguments, an integer file descriptor for the parent directory, a character string for the file pathname for the file to unlink, and integer that is for all of the flags for the function. It returns an integer 0 on success, and a -1 on error.