1. Overview

In this lab you will be introduced to a simple Unix systems programming problem, namely printing out statistics about files. We present the program in the form of a Unix `man`(1) page. You will also need to read `man` pages as listed.

NAME
> l9stat — stat a list of files

SYNOPSIS
> `l9stat` [*filename ...*]

DESCRIPTION
> Each argument is a filename whose statistics are to be printed. If no file-names are given, the current directory (.) is used. For each file, print the mode, the file size in bytes , the file or link modification time, and the name. If a file is a symbolic link, the value of the link is also printed. If the time is more than 180 days from the current time, the month, day, and year are printed, otherwise, the month, day, and time.

EXIT STATUS
> 0   No errors were detected.
>
> 1   Errors were detected and messages printed.

SEE ALSO
> `ls`(1), `man`(1), `stat`(1), `lstat`(2), `readlink`(2), `time`(2), `localtime`(3), `strerror`(3), `strftime`(3).

2. Notes

To prepare for writing this lab, read several `man`(1) pages which describe the system calls and library functions you need to use. The number in parentheses indicates the chapter : (1) User commands, (2) System calls, and (3) C library functions.

(α) The commands `ls`(1) and `stat`(1) show information about files. The Perl program `l9stat.perl` is a reference implementation, and your program should produce the same output.

(β) Manual pages can be read using the `man`(1) command. Each man page is in a particular section, so, for example, the command

```
man -s 3 printf
```

can display the man page for `printf`(3). The number in parentheses indicates the manual section.

(γ) You may use the `x` command from **/afs/cats.ucsc.edu/courses/cmps012b-wm/ bin** (in your path) to display a man page in a new **xterm** :

```
x man -s 3 printf &
```

(δ) The system call `lstat`(2) provides information about a file (or a link, if it is a link). The information is in a `struct stat`, and you will use the fields `st_mode`, `st_size`, and `st_mtime`. The mode is printed as 6 octal digits, and the size as 9

decimal digits.

(ε) If **lstat** fails, print an error message in the standard format and continue.

(ζ) The system call **readlink**(2) will return information about the symbolic link, if the file is a symbolic link. If it fails, assume that the file is not a symlink and ignore the failure.

(η) The system call **time**(2) returns the current time in seconds since the Epoch (Thu Jan 1 00:00:00 1970 UTC) as a value of type **time_t**.

(θ) The C library function **localtime**(3) accepts a value of type **time_t**, and using the local time, formats it into a **struct tm** suitable for printing.

(ι) The C library function **strerror**(3) accepts the variable **errno** and returns a suitable formatted string indicating a reason for a failed system call.

(κ) The C library function **strftime**(3) accepts a format string and a **struct tm** and returns a string suitable for printing. Use the format string "**%b␣%e␣%R**" if the time is within 180 days of the current time (whether in the past or in the future) and "**%b␣%e␣␣%Y**" if beyond that time. The symbol "␣" is a visible space, which you should type into your program as an ordinary space.

3. What to submit

Submit **README, Makefile, l9stat.c.** Your **Makefile** must build an executable binary called **l9stat**. Also, if you are doing pair programming, submit the required pair programming files.