ar key archive file file1 file2 ~~~

q

t

Magic #

Arc Header

file content

Arc Header

file content

Can only use read/write/(seek)

CS 551: Assignment 1

Doing Unix File I/O

Due Monday September ~~26~~ 30 (Friday), 2016 at 10PM

Write a C program on Unix called "myar.c". This program will illustrate the use of file I/O on Unix, by maintaining a Unix "archive" library, in the standard archive format.

Once compiled your program should run in a manner similar to the standard Unix command "ar". You may wish to look at the Unix man page for "ar" for insight, however, for this assignment the following is the syntax your program must support:

myar key afile name ...

where afile is the name of the archive file to be used, and key is one of the following options:

q quickly append named files to archive

✓ x extract named files

✓ xo extract named files restoring mtime

✓ t print a concise table of contents of the archive → names

✓ tv print a verbose table of contents of the archive → names & meta info

d delete named files from archive

A quickly append all "regular" files in the current directory (except the archive itself)

lseek helps to skip contents ~~just~~ only hit headers

The archive file maintained must use exactly the standard format defined in "/usr/include/ar.h", and in fact may be tested with archives created with the "ar" command. The options listed above are compatible with the options having the same name in the "ar" command The "A" command is a new option not in the usual "ar" command.

rwx

4 2 1

0 0 0

1 1

Notes:

(1) You may assume all files are in the current directory, no filename is longer than 15 bytes, and no archive index will be attempted. You MUST NOT consult source code for other implementations of "ar".

(2) For the "q" command "myar" should create an archive file if it doesn't exist, using permissions "666". For the other commands "myar" reports an error if the archive doesn't exist, or has the wrong format.

(3) You will have to use the system calls "stat" and "utime" to properly deal with extracting and restoring the proper timestamps. Since the archive format only allows one timestamp, store the "mtime" and use it to restore both the "atime" and "mtime" if the "o" option is specificed. Permissions should also be restored to the original value, subject to "umask" limitation. → record, more detail next time.

(4) The "q" and "A" commands do not check to see if a file by the chosen name already exists. It simply appends the files to the end of the archive.

(5) The "A" command saves only ordinary files and symbolic links. In the case of a symbolic link it saves the path to the ultimate destination, NOT the contents of the destination. This is different than what "q" does.

(6) The "x" and "d" commands operate on the first file matched in the archive, without checking for further matches. → record,

(7) In the case of the "d" option, you will have to build a new archive file to recover the space. Do this by unlinking the original file after it is opened, and creating a new archive with the original name. ← remove command

(8) You are required to handle multiple file names as arguments.

(9) Since file I/O is expensive, do not make more than one pass through the archive file, an issue especially relevant to the multiple delete case.

if the size is even, arc header continue

If the size is odd, skip 1 byte and continue arc header