SY204 – Sys Prgm & OS Fund

Name(s): ______________________

Alpha(s): ______________________

Assignment Type:	Programming – Laboratory	Collaboration Policy:	Default
Assignment Title:	0x03: Untouchables		

1. General.

Due	
Portion of Assignment	Course Server Time / Date
Pseudo Code	Friday, 1800 03 Mar 2017
Complete	Beginning of Class Tuesday, 07 Mar 2017

a. Description. Read through this entire assignment before starting work. In this assignment you will create a program that modifies the contents of a file but after the file has been modified the access time and modified time metadata will be the same as before the file was modified.

2. Explore (15).

a. (8) Read the `stat(2)` man page, and answer the following questions.

[1 / 0] `stat`ing a file modifies the last accessed time field? True / False

[1 / 0] `stat`ing a file modifies the last modified time field? True / False

[1 / 0] `stat`ing a file modifies the last status change time field? True / False

[2 / __ / 0] The `st_mode` field contains what two types of information about an entity in the file system?

[1 / 0] What `open(2)` option does not alter the last accessed time field? ____________

[1 / 0] What defined constant is a bit mask for file type information? ____________

[1 / 0] The below code is equivalent to what POSIX defined macro? ____________

```
if ( stFile.st_mode & S_IFMT == S_IFIFO ) {
    return 1;
} else {
    return 0;
}
```

Name(s): ______________________

Alpha(s): ______________________

b. (4) Read the `futimens(3)` man page, you will likely need to review the `stat(2)` man page as well, and answer the following questions. Given the below variable declarations:

```
struct stat fInfo;    struct timespec times[2];
```

[2 / 0] The `times[0].tv_sec` field is equivalent to what `fInfo` field?

[2 / 0] The `fInfo.st_mtim` field is equivalent to what `times` element?

c. [3 / __ / 0] Based on the topics covered in this assignment, it is clear that the last modified time cannot be used to determine if the contents of a file were changed. Describe what would need to be done to determine if the contents of a file had changed, recall topics from:
SY110 – Crypto activities `http://rona.academy.usna.edu/~sy110/index.html`
SY201 – Crypto activities
`http://yog.academy.usna.edu/~hoffmeis/teaching/sy201/index.html`

3. Plan (20). You may attach additional pages as needed.

a. [4 / __ / 0] List the systems calls or library routines, beyond `printf(3)` family, you expect to use in the completion of this assignment, and describe what you will use that system call for; `open` is provided as an example, list at least four more.

System Call	Use Description
`open`	Request access to a file within the file system.

Name(s): ____________________
Alpha(s): ____________________

b. [8 / __ / 0] Draw a high-level block control flow diagram that depicts the order in which you expect to use the system calls and library routines that you discussed in Paragraph 3.a.

c. [4 / __ / 0] Based on your block control flow diagram, write C code to include the header files required to call the system calls and library routines discussed in Paragraph 3.a. Write the C code in `untouchPseudo.c` . Include a comment that clearly indicates your code labeled *Paragraph 3.c*.

d. [4 / __ / 0] Based on your block control flow diagram, write C code to declare the variables you will need to pass as arguments to the system calls and library routines discussed in Paragraph 3.a. in your `main` function. Write the C code in `untouchPseudo.c` . Include a comment that clearly indicates your code labeled *Paragraph 3.d*.

4. Implement (35).

a. [0 / -5 / -10] Sound and Secure Cyber Design

- You shall minimize the use of global variables other than `errno`, `opt*`, and for error reporting.
- You are encouraged to use `#define`, and other preprocessing directives.
  - Your program will not be compiled using the `-d` compiler directive.
- Source code is poorly or inconsistently formatted, but can be compiled (readability).
- Source code uses poorly named functions or poorly named variables (readability).
- Source code does not use associated common libraries (reusability).
- Source code does not use library defined constants (portability).
- Source code does not use deprecated language features (design for the future).
- Implementation does not violate Principle of Least Privilege.

b. [3 / __ / 0] Usage Option (All Parts)

- Read in the "`-h`" (help) option using `getopt(3)`, that does not take any arguments.
- Upon detection of the "`-h`" option display simple usage information that describes how to use your program. Example: program name, listing and discussion of command line arguments.
- You may assume that all options will appear before arguments.
- If no command line arguments are passed in, output the usage message.
- You may use Stream I/O (`printf(3)`) for this feature.
- After you output the usage message, exit the program with a normal return value.
- Implement the outputting of the usage message as a function your program calls.

c. [2 / __ / 0] Error Handling (All Parts)

- For an error that occurs from a system call, output a simple error message (recall `perror(3)`), and exit the program returning an error number.
- At least return a different error number for each of the systems calls that you use. That is, multiple calls to the same system call need not return unique error numbers, but calls to different system calls shall return unique error numbers.
- You may use Stream I/O for reporting errors.
- Output error messages to standard error.

d. [10 / __ / 0] Part 1 – These Aren't The Time Stamps You're Looking For

- Open the regular file passed in via the command line argument and append at least one printable, non-whitespace character to the file.
- After you have appended data to the file, ensure the original last access time and last modified times are restored to at least the second.
- You may assume the command line argument is either in the current working directory, or is not in the file system (there is no path section to the command line argument).
- If the command line argument does not exist, exit the program returning a unique error number.
- Do not assume the command line argument is a regular file. You may assume that if the command line argument is a regular file the file will be an ASCII text file.
- If the command line argument is not a regular file, output a simple error message and exit the program returning a unique error number.

e. [10 / __ / 0] Part 2 – Time Is On My Side

- Continuing from Part 1, after you have appended the data to the file, ensure the original last access time and last modified times are restored to at least the nanosecond.

f. [10 / __ / 0] Part 3 – I've Been Here Along

- Complete all of Part 2 before beginning Part 3.
- If the command line argument does not exist, then create a regular file named as the command line argument.
- Write at least one printable, non-whitespace character to the file.
- The created file shall be readable and writeable by the user.
- The created file shall have the same last access time and last modified time as the parent directory.

Name(s): ___________________________________

Alpha(s): ___________________________________

g. [+ 5 / 0] Part 4 – Extra Credit

- Complete all of Part 3 before beginning Part 4.
- Do not assume the command line argument is in the current working directory (see `basename(3)`).

5. Test (30).

a. [0 / -15] Code Compilation and Running

- Source code compiles (`gcc -Wall ...`) under Ubuntu 16.04 LTS x86-64 with no warnings and no errors.
- Compiled program runs without crashing on test cases (no SEGFAULTs, no infinite loops).

b. [3 / __ / 0] Usage Option (All Parts)

- Solution correctly outputs a usage message when the help option is entered.
- Solution correctly outputs a usage message when no command line arguments are entered.
- Solution correctly returns a normal exit value.

c. [3 / __ / 0] Error Handling (All Parts)

- Solution correctly outputs error messages to standard error.
- Solution correctly returns unique values for different system calls.

d. [8 / __ / 0] Part 1 – These Aren't The Time Stamps You're Looking For

- Solution modifies contents of file.
- Solution leaves access time unmodified to at least the second.
- Solution leaves modified time unmodified to at least the second.

e. [8 / __ / 0] Part 2 – Time Is On My Side

- Solution modifies contents of file.
- Solution leaves access time unmodified to at least the nanosecond.
- Solution leaves modified time unmodified to at least the nanosecond.

f. [8 / __ / 0] Part 3 – I've Been Here Along

- Solution creates file.
- Solution mirrors access time of parent directory to created file.
- Solution mirrors modified time of parent directory to created file.

SY204 – Sys Prgm & OS Fund Name(s): ____________________
Alpha(s): ____________________

6. Submit. Turn in this worksheet and submit source code per the following specifications.

- Turn in this worksheet to your instructor after you have submitted your final source code.
- Submit your pseudo code as a single file named `untouchPseudo.c`, following the submission directions on the course information web page.
- Submit your final code as a single file named `untouch-#.c`, where `#` is the part of the implementation that you completed (e.g. `untouch-4.c`), following the submission directions on the course information web page.
- `submit` Assignment (project): *lab03*

[0 / -10 / -15 / -20] Submitted source code that demonstrated insufficient effort (lack of attempt to solve problem)
[0 / -5] Submitted assignment using incorrect assignment type or incorrect assignment number.
[0 / -5] Submitted incorrect number of files in submission directory.
[0 / -5] Submitted file(s) named incorrectly.
[0 / -5] Submitted file(s) do not include name and alpha.