C Review 2310 Notes  
  
  
Tar:  
Often called a tarball. Used to create an archive of a set of files. The gz part compresses the files.

Syntax:  
 tar<flags><name of the tar.gz><list or path to the files>

ORDER MATTERS

Flags (order does not matter):

c: Create an archive.

z: Compress the archive with gzip.

v: Display progress in the terminal, also known as “verbose” mode.

f: Allows you to specify the filename of the archive.

DO NOT TURN IN A CORRUPT FILE! YOU WILL GET A ZERO!

School of computing ssh stores backups:

ls -la > scroll until green, highlighted backup folder called .snapshot > Folder desired (hourly, daily, or weekly).

Change the c flag to an x flag in order to extract/open a tarred file  
  
-C is used to untar to a specific location.

Header Guards (only in .h files):

#ifndef UNIQUE\_NAME

#define UNIQUE\_NAME

CODE

#endif

File Handling

Create

Open

Edit

Close

Types of files:

Text-based files:

Readable

Editable

Binary Files:

Can hold higher amount of data.

Not easily readable

More secure

Opening a file:

(declare a pointer)

FILE\* ptr;

ptr = fopen(“filename”, “mode”)

Can also do both at once: FILE\* ptr = fopen(“filename”, “mode”)

Assertions:

Used to test assumptions (Evaluates to True or False)

FILE\* input = fopen(“file.txt”, “r”);

^^^ This section of the fopen command is the mode

assert(input != NULL);

^^^ This is the name of the file pointer

This will abort the program if the expression evaluates to false (0).

Need to include the <assert.h> library.

Reminder:

cin = scanf (user input) / fscanf (getting data from a file)

cout = printf

Rewind and append:

Function for file pointers. This takes the pointer from whatever it is pointing to and brings it to the beginning of the file. Append will ALWAYS add to the end of the file; it does not matter where the pointer is pointing to in the file.

size\_t data type:

Is an unsigned intergral data type

Defined in several C libraries

Represents object sizes and is guaranteed to be big enough to contain the size of the biggest object the system can handle.

Used by many system calls: malloc, calloc, memcpy, strlen, etc.

Portable, Unsigned (can’t be negative), Performance , Standardization

fread():

size\_t fread(void \* ptr, size\_t size, size\_t n, FILE\* fp);

^^ Data stored here. ^^ Data read from

size\_t size = the size of each “n” item read in

size\_T n = the number of items to be read

Returns n on success and less than n on error or eof.

Fwrite():

Size\_t fwrite(const void \* buffer, size\_t size, size\_t n, FILE\* fp);

The end arguments are the opposite of fread.