**Notes for Test 2**

1. **Environment Variable:** these variables are used to customize the operating system and the shell (use SETENV or EXPORT)
2. **Positional Parameters**: command line parameters (using $1, $2, etc.)
3. **Session Variables:** defined by the OS or Web Server when the user logs in
4. $# : number of arguments

$-: options supplied to the shell

$? : exit value

$$ : process number

$! : process number of the last command done in background

$n : where n is from 1 through 9, reading left to right

$0 : the name of the current shell or program

$\* : all arguments on the command line as a string

$@ : all arguments on the command line as an array

1. **Methods of passing data to a script:** command-line, user input, pre-initialized within the shell memory
2. **PATH:** is used by the shell when you type ./script or ./program
3. **CLASSPATH:** is used by Java when you run a Java program
4. **DATAPATH**: the path to import data
5. **Master backup script**: in the HOME directory, a script listing all the files to backup
6. **Project specific backup script:** Similar to the master backup script, but contained in the project’s top directory
7. **System scripts:** Boot scripts, login scripts, logout scripts
8. **Session & Environment Variables contain information about:** your IP address, the shell you are using, your username
9. **You can use the “which” command to:** figure out what file path is needed.
10. **A Session:** the run-time environment or the Shell Environment.
11. **du [options] [directory or file]:** report amount of disk space in use
12. \r Carriage return

\t Tab

[\\ Backslash](file:///\\-Backslash)

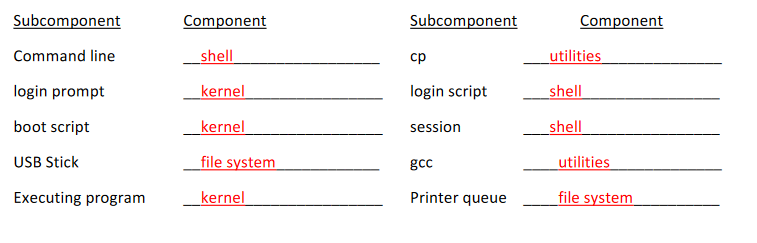
\a Bell

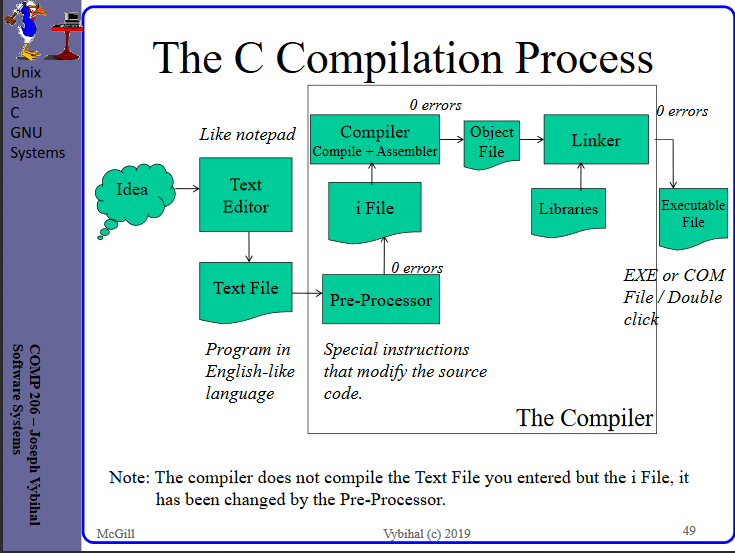
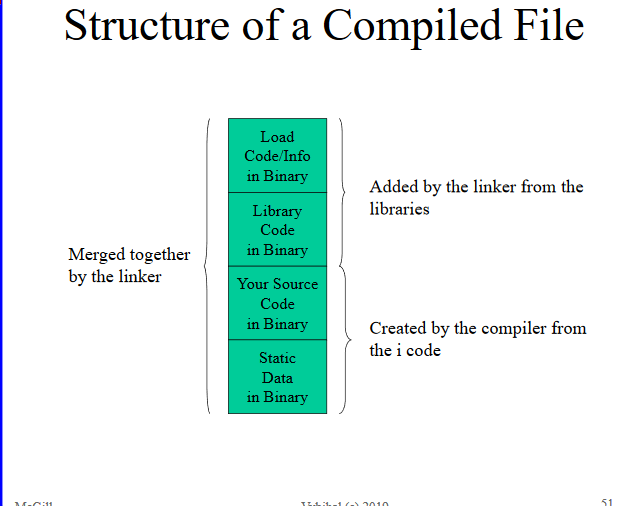
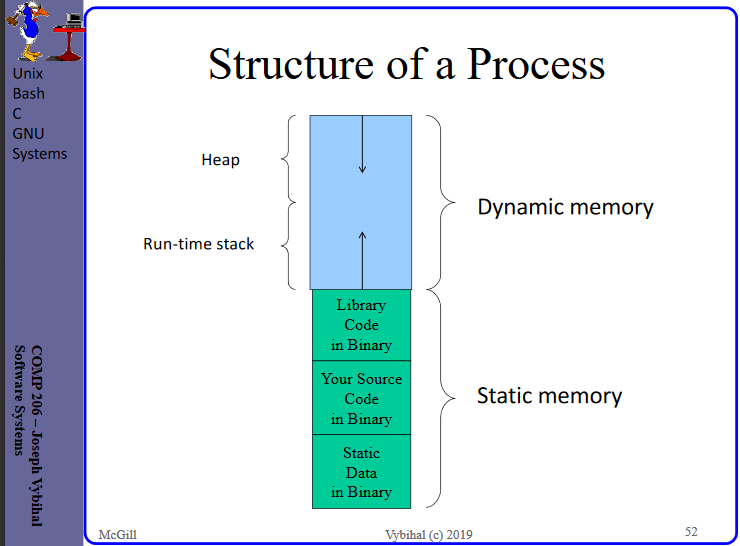
\b Backspace (without delete)

1. **Pre-processed File:** FILENAME.i

**Object Files and Assembler Files:** FILENAME.o, FILENAME.s

1. **Three types of I/0:** Console, Stream, Files
2. **Console I/O:** Input and output - keyboard and screen.
3. **Stream I/O:** consumes n bytes of data in a continuous sequence over time.
4. **File I/O:** Reading and writing to a file on disk
5. **Three standard streams exist in C:** stdin, stdout, stderr
6. **The symbol 2>:** redirects stderr, or the second standard output stream, to the file



1. **Difference between compiler and interpreter:** Unlike an interpreter, a compiled program can speak directly to the CPU (additional speed)
2. **A compiler:** attempts to convert your source code to machine code.
3. 
4. 
5. 
6. **Strcpy:** strcpy(array, “first words”)
7. **Strcat:** strcat(array, “second words”)
8. **Strncmp:** strncmp(“mark”, “mary”, 3); // 0
9. **Memset:** memset(array2, ‘\*’, 50); //put 50 asterisks in the array
10. **2-pass compiler:** scans the source file twice
11. **1-pass compiler:** scans the source file once
12. **Conclusion:** 1-pass is faster than 2-pass
13. **C is a:** 1-pass compiler
14. **Restriction:** 1-pass, by definition, has a declaration restriction (declaring function after main is bad)
15. **First come first serve server scope rule:** block, local space, global space, external space, syntax error

Programming notes:

1. In printf%s prints until CR. In scanf%s reads until CR or space
2. fgets() can read from any open file, but scanf() only reads standard input. fgets() reads 'a line of text' from a file; scanf() can be used for that but also handles conversions from string to built in numeric types.
3. For sprint, the output goes to CHAR\_ARRAY. For sscanf, the input comes from CHAR\_ARRAY
4. int main(int argc, char \*argv[]) {
5. include<ctype.h>
6. Case manipulation:

Int c = toupper(int);

Int c = tolower(int);

Character testing:

Int x = isalpha(int);

Int x = isalphanum(int);

Int x = isdigit(int);

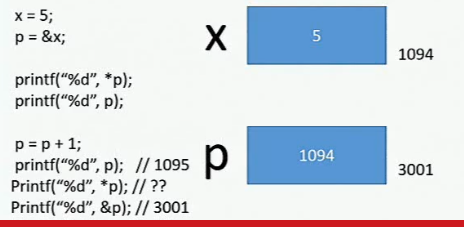
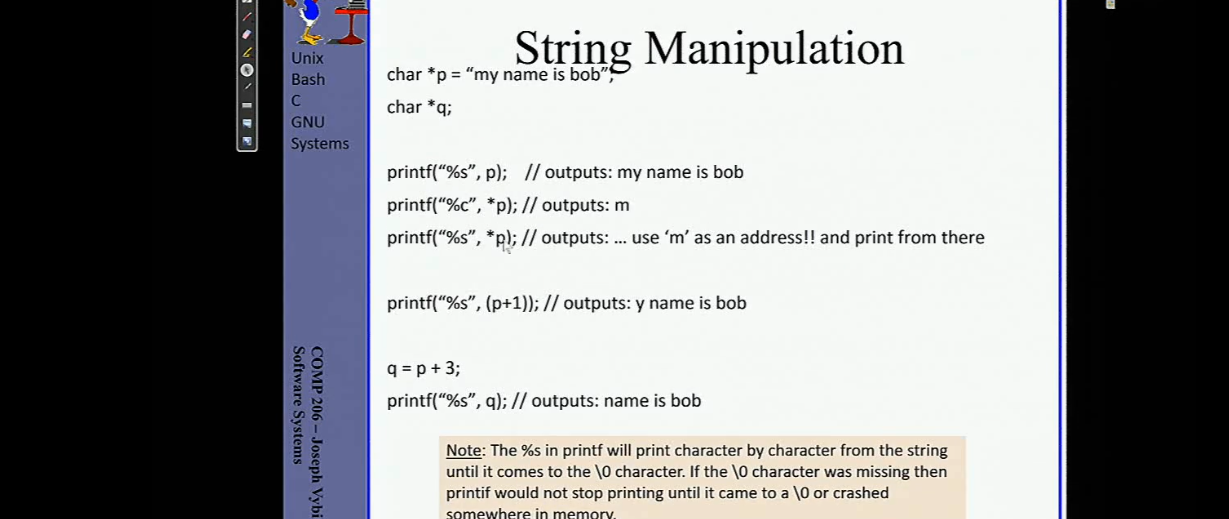
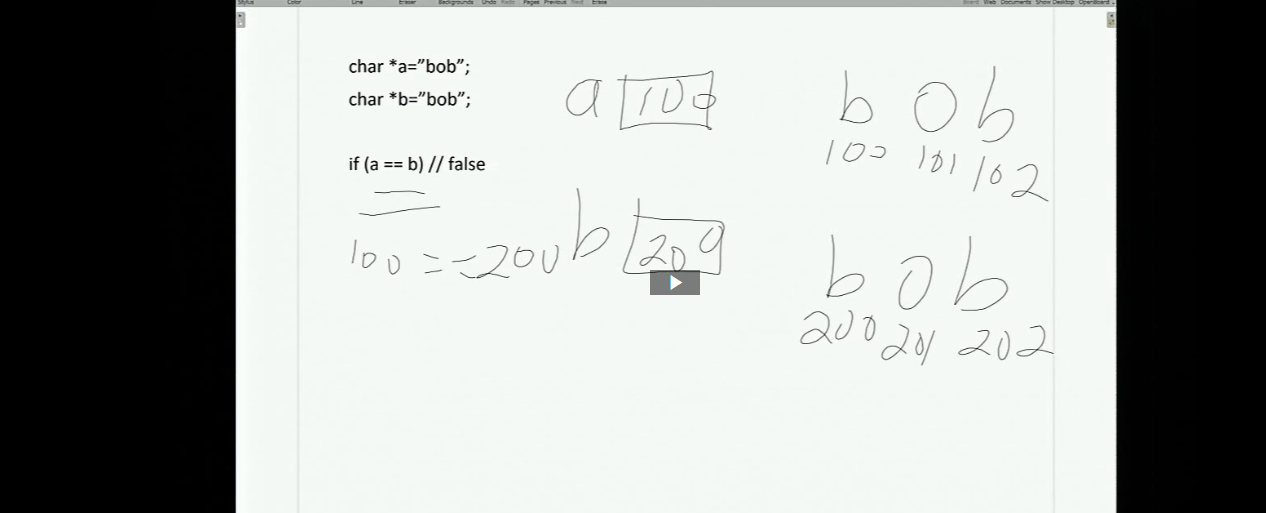
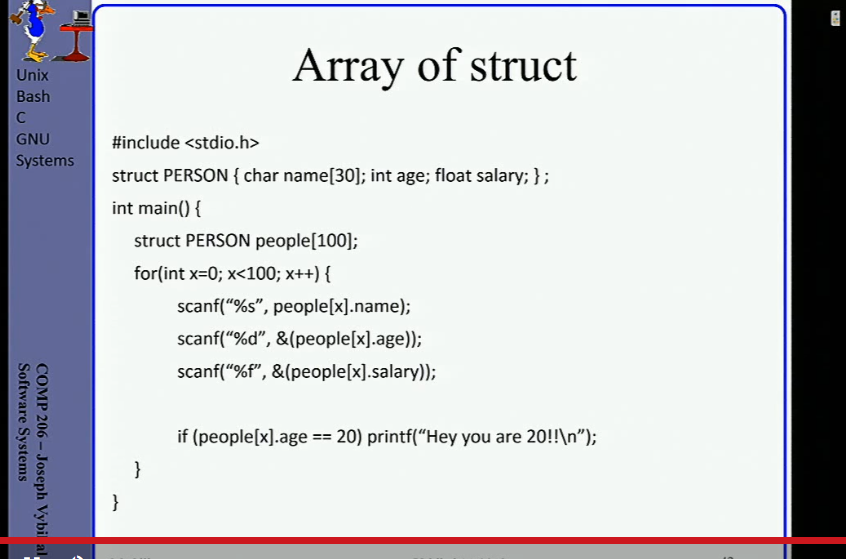
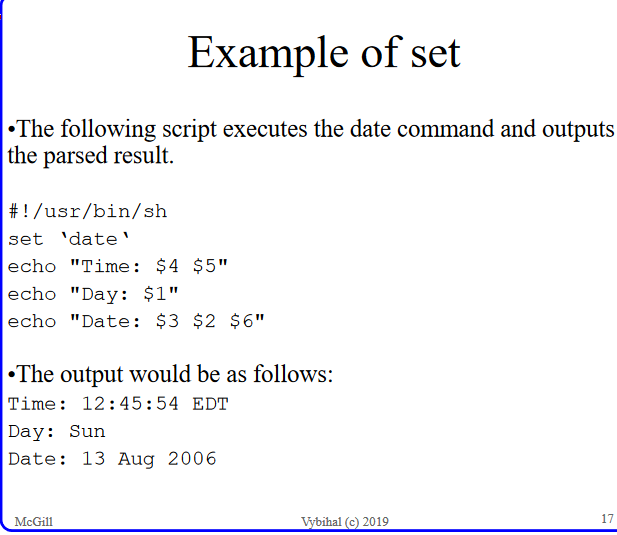
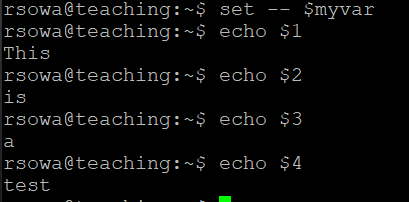
If not in path, will look in classpath

1. “#” is pre-processor
2. “puts” returns an error code value
3. “Stream” can behave like “Console”
4. “getchar” has no argument, does not use stream
5. “getchar” will take many letters, but will only take the first
6. “Stream”: “river of characters”
7. fgets(array,limit,stream)
8. “%5d” allocated 5 slots, “%-5d” allocates 5 slots and puts number at beginning
9. “%5.1” \_ \_ \_ . \_
10. scanf reads till space or carriage return
11. fgets gets everything
12. Don’t need “&” for array
13. “gets” stops when it sees a new line
14. Use “system” for commands
15. int x = 5;

int \*p;

p = &x;

printf(“%d”, p); prints the address of x

1. 
2. 
3. 
4. “Strcpy” only works with arrays destination
5. Can move pointer, but can’t move array
6. 
7. struct Person a,b,c
8. 
9. “printenv” to view all environment variables
10. 
11. Use “&” for ints