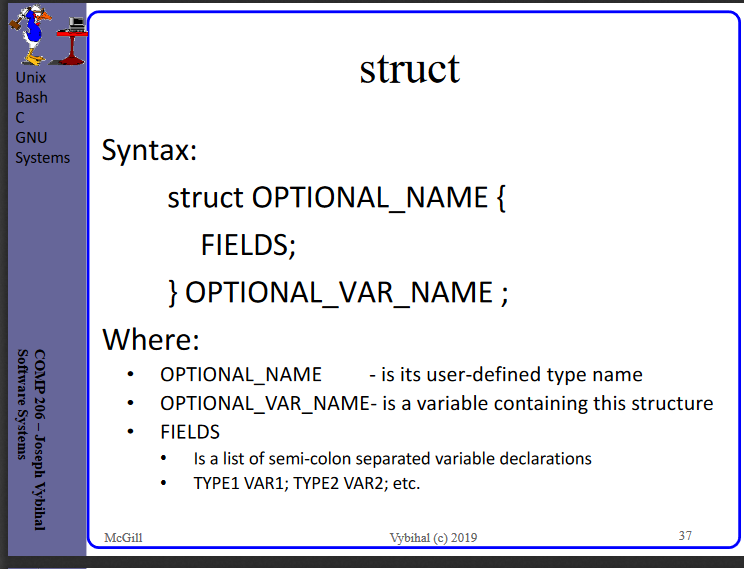
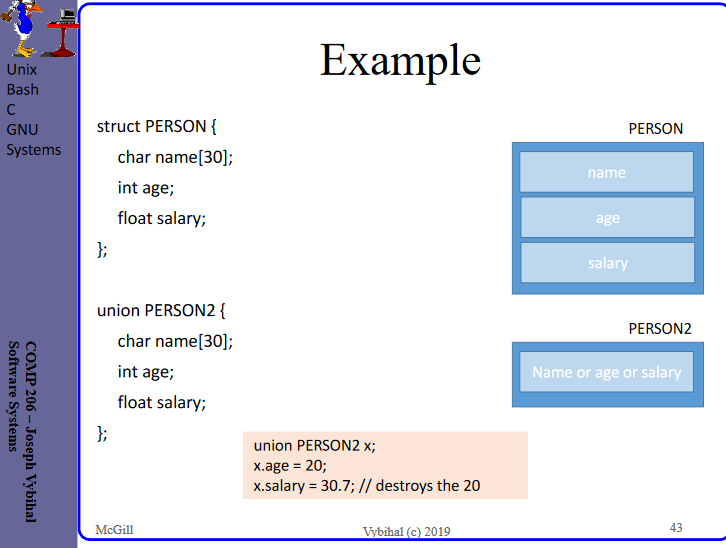
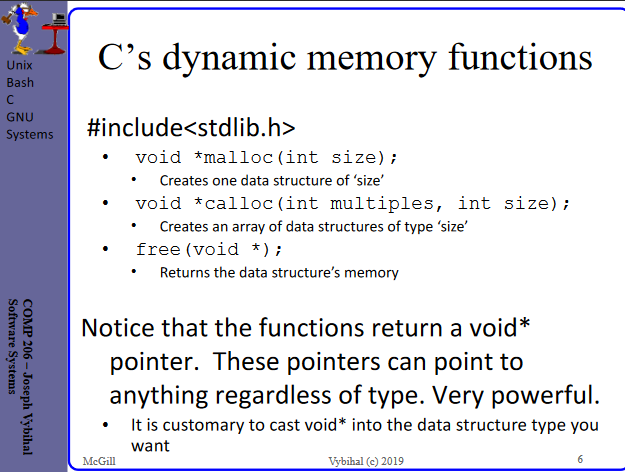
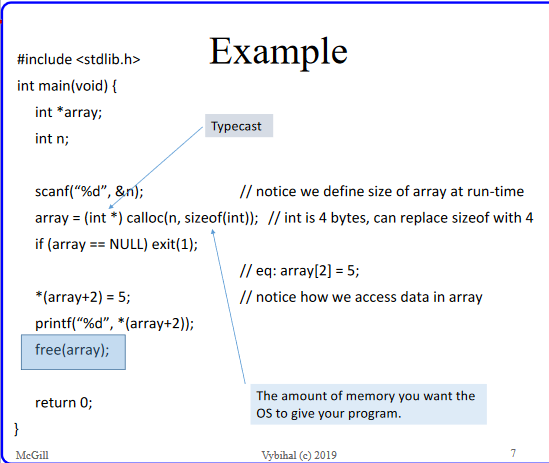
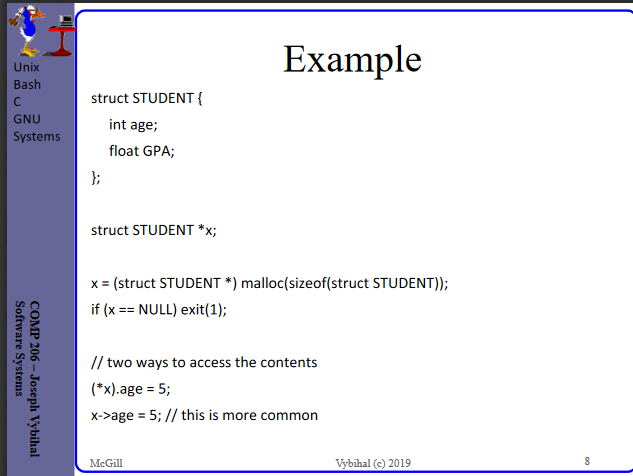
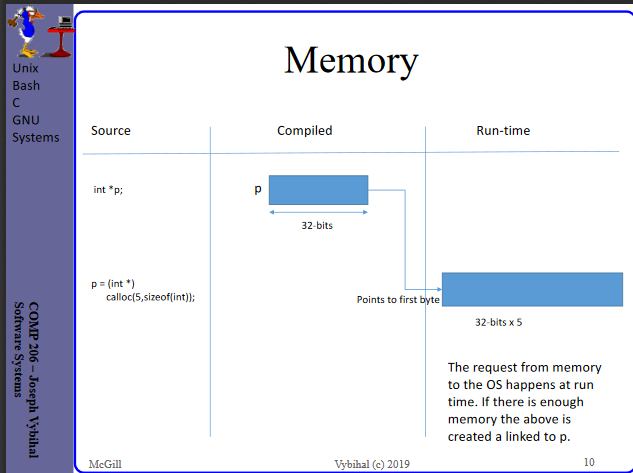
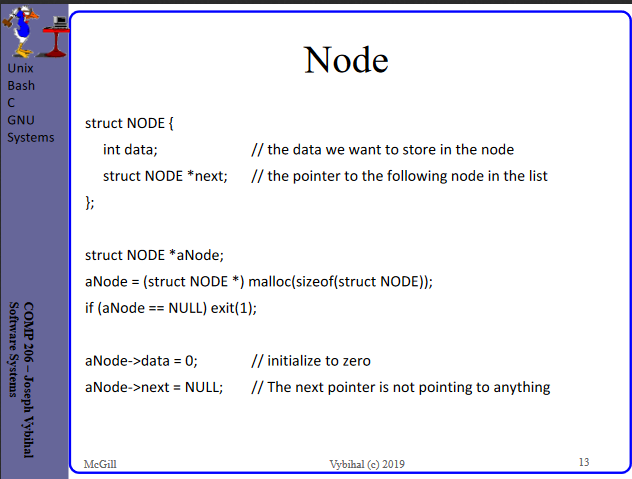
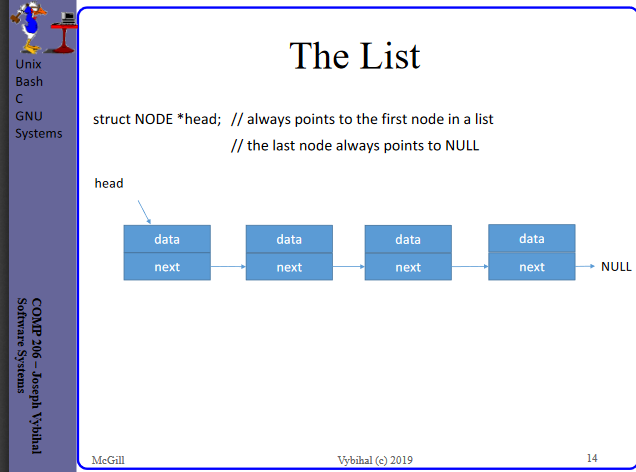
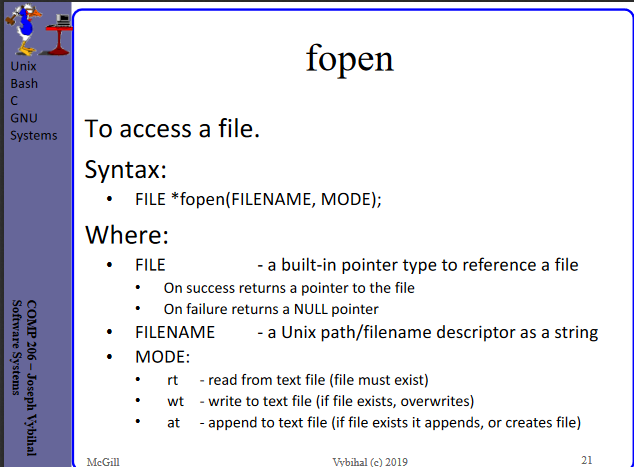
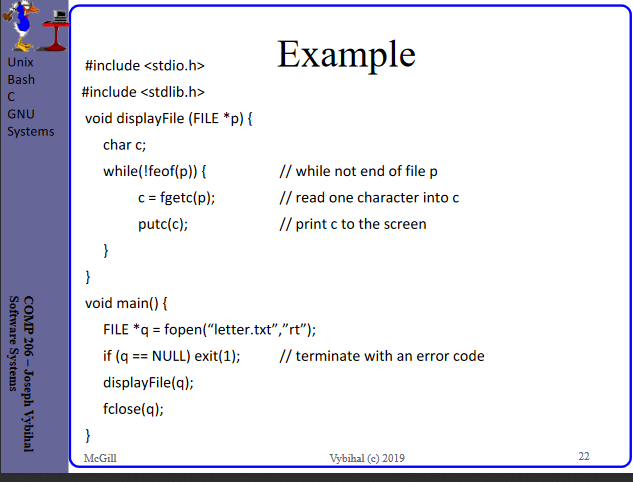
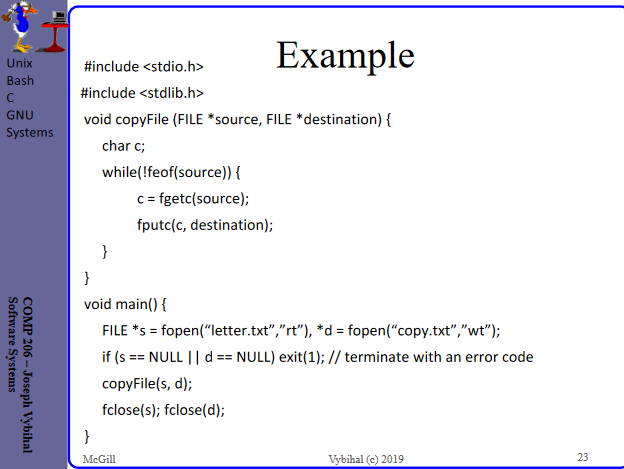
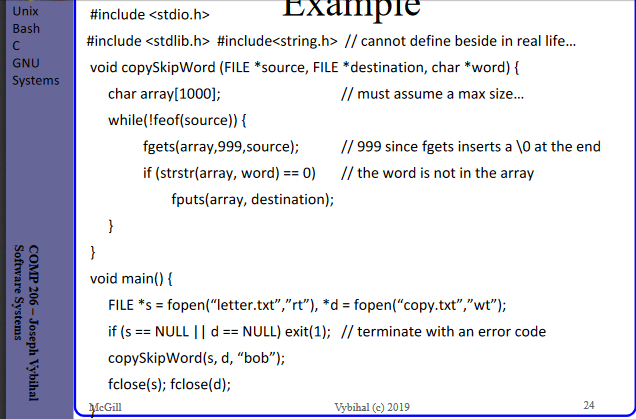
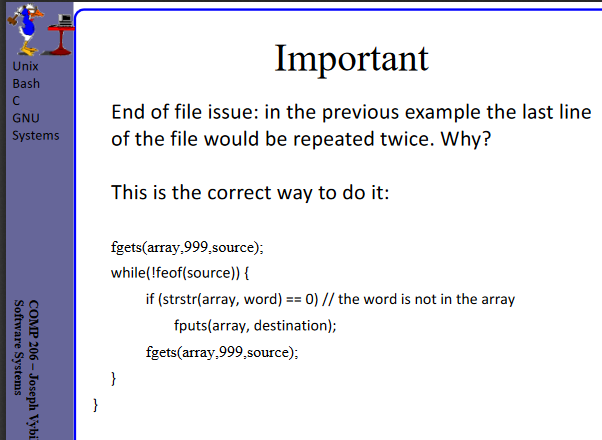
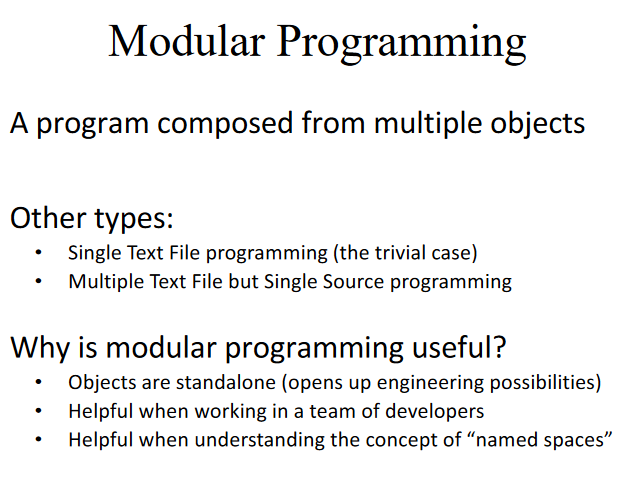
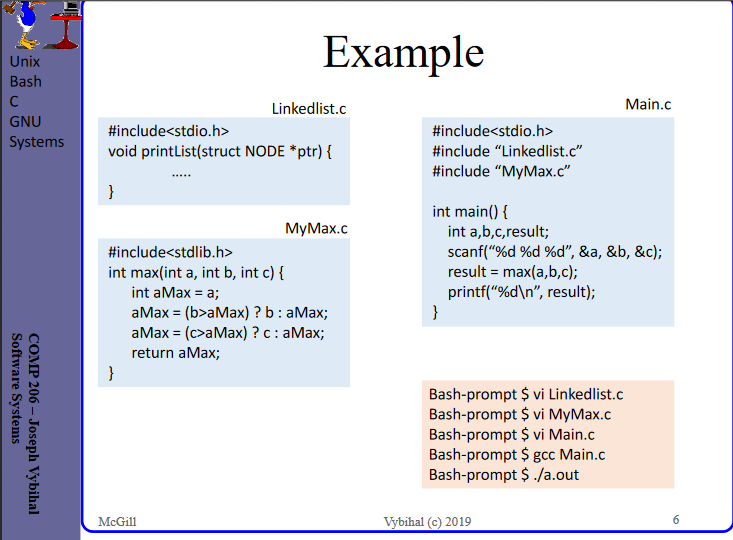
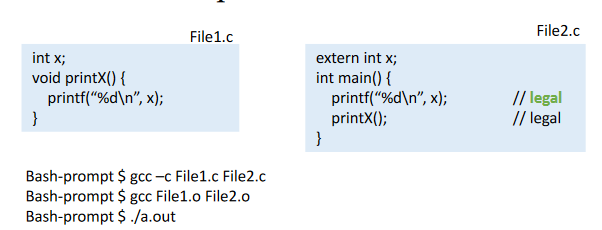
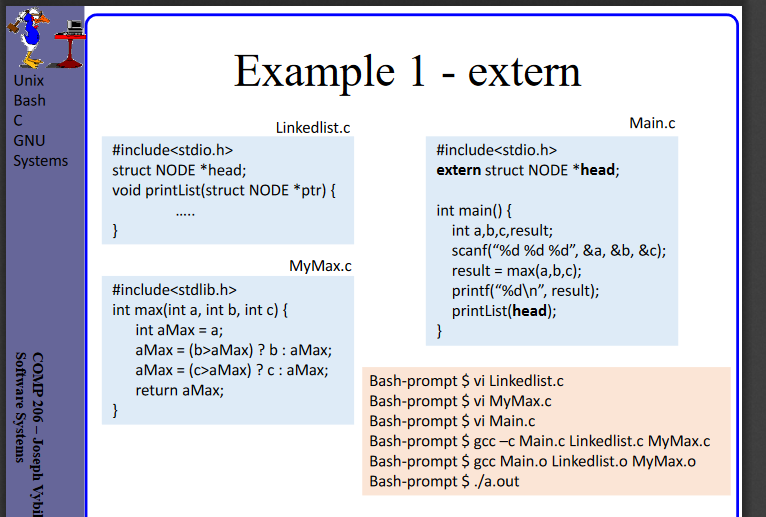
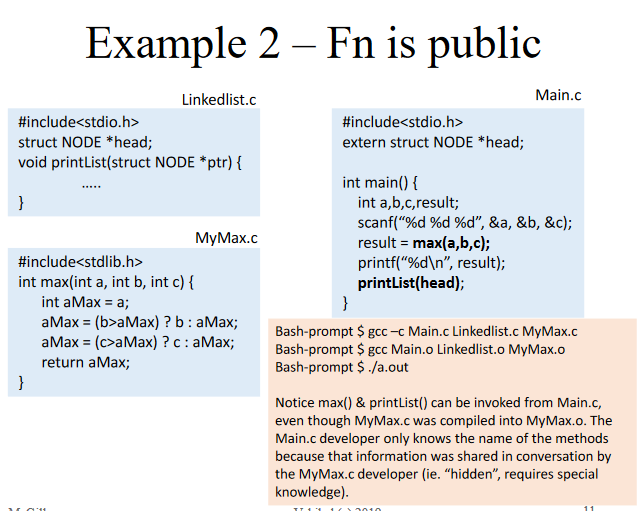
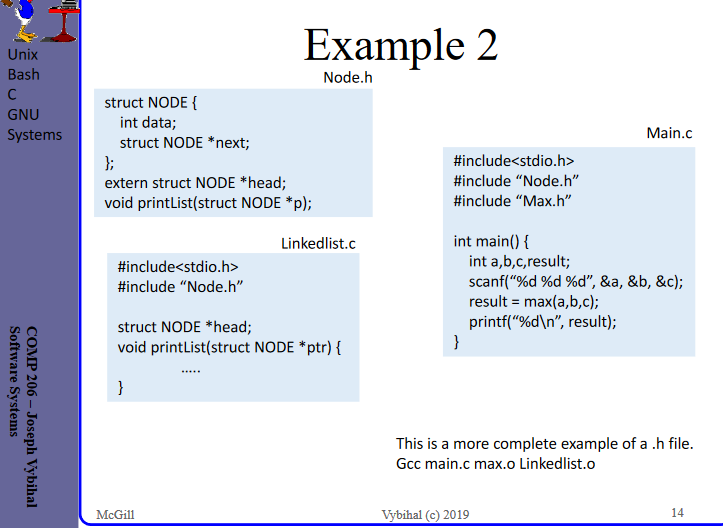
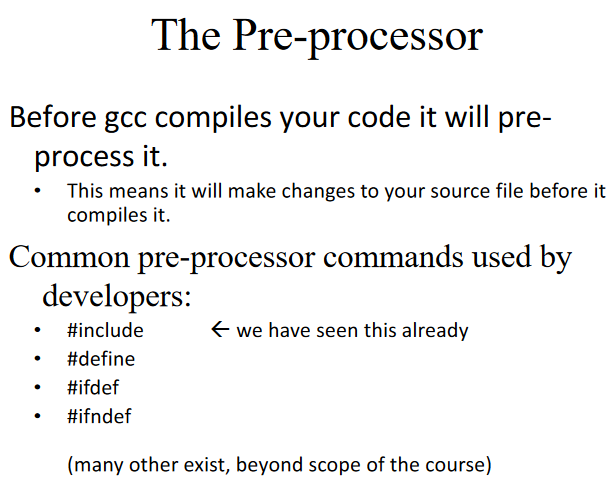
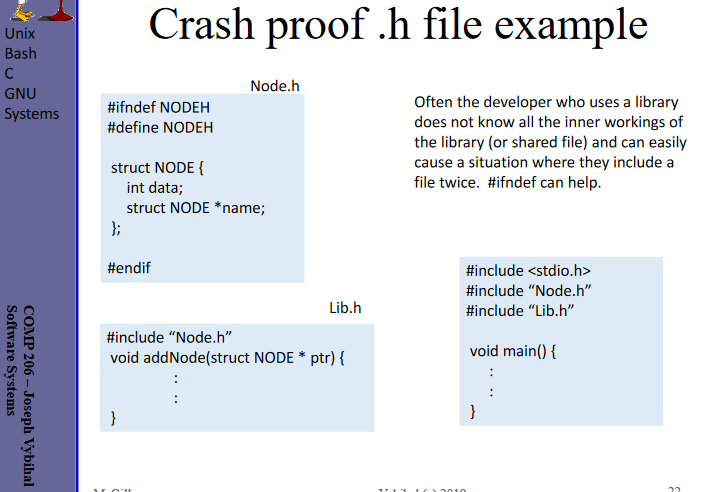
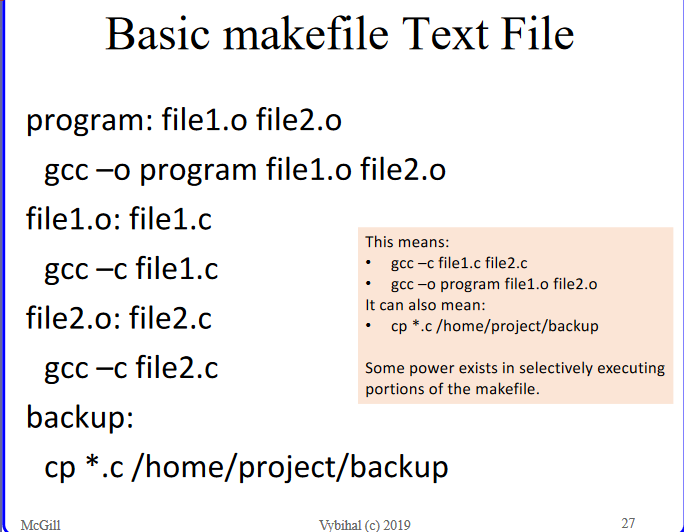
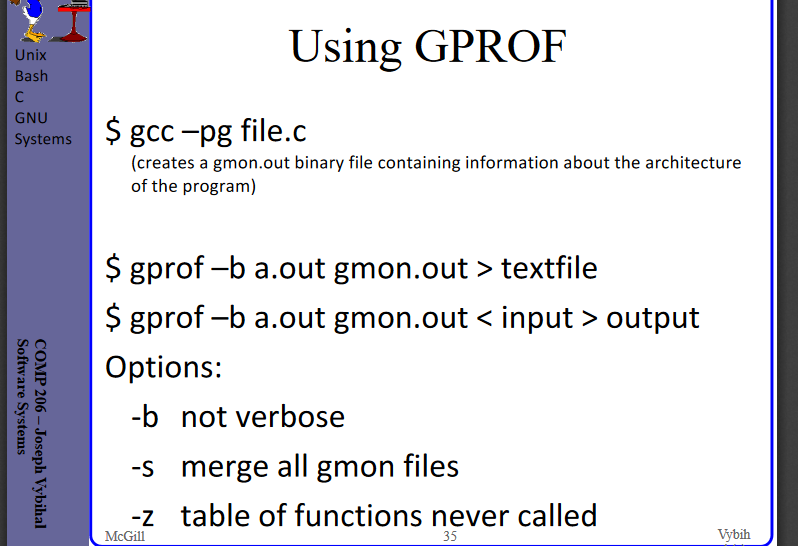
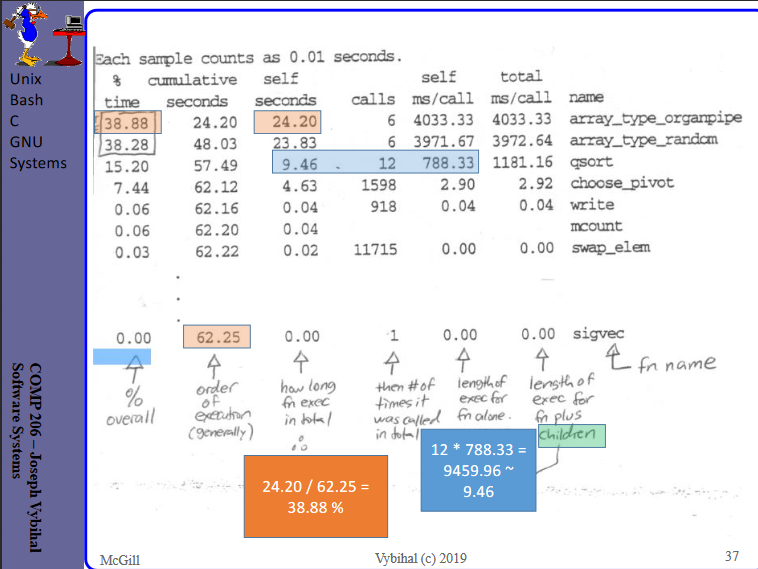
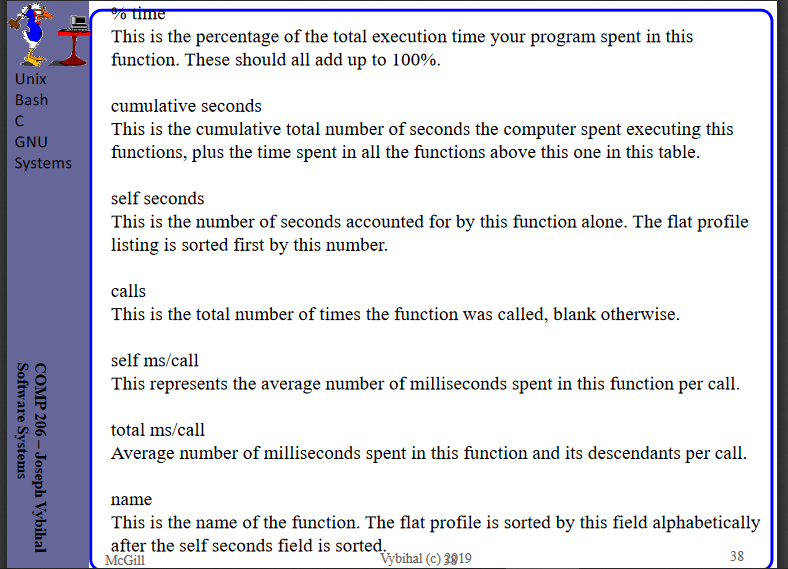
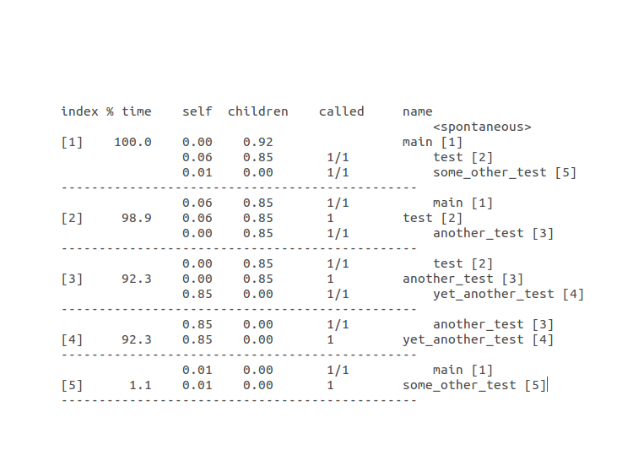
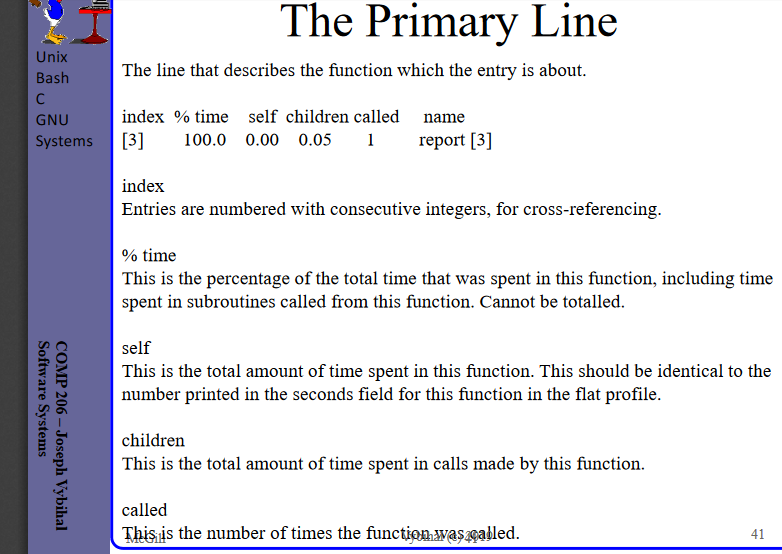
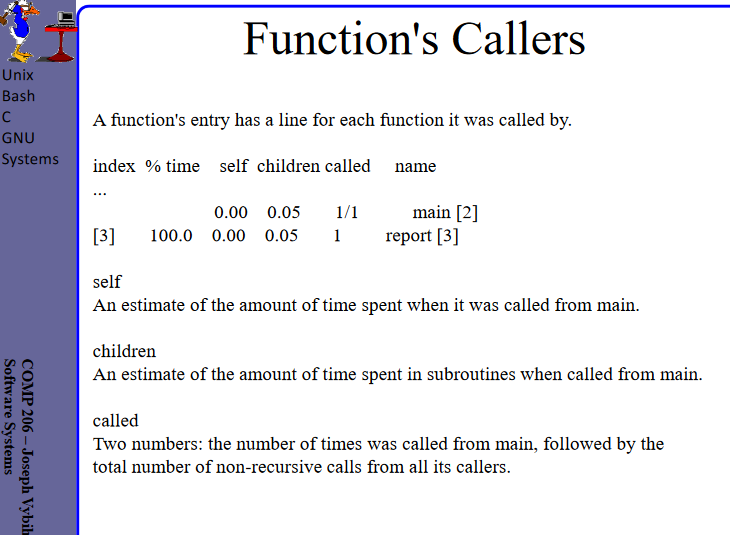
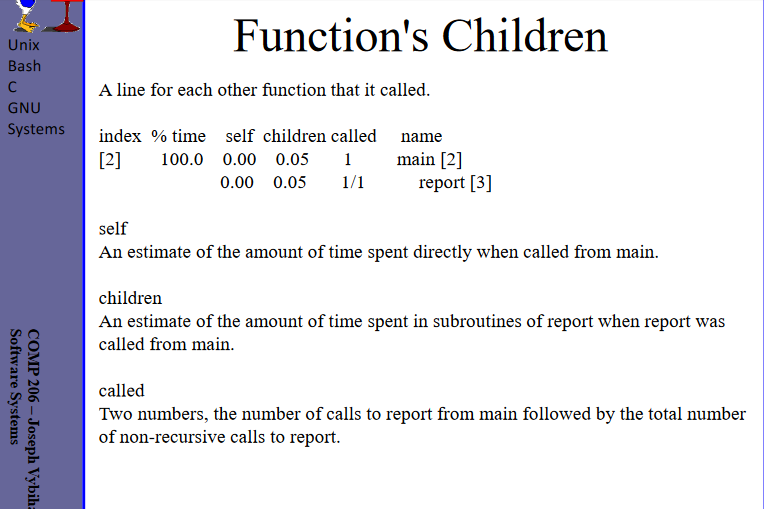
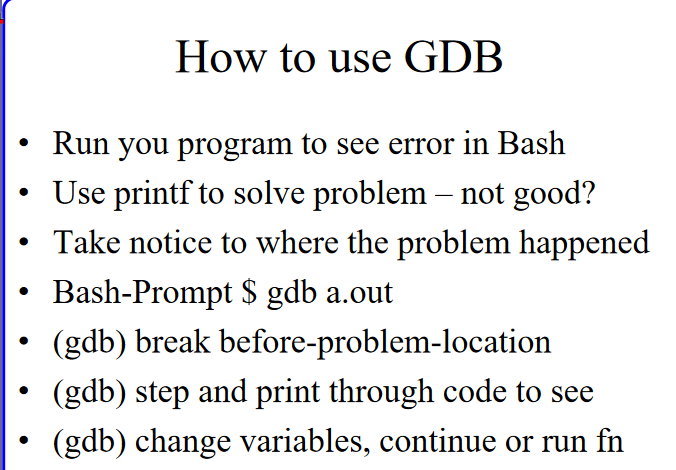
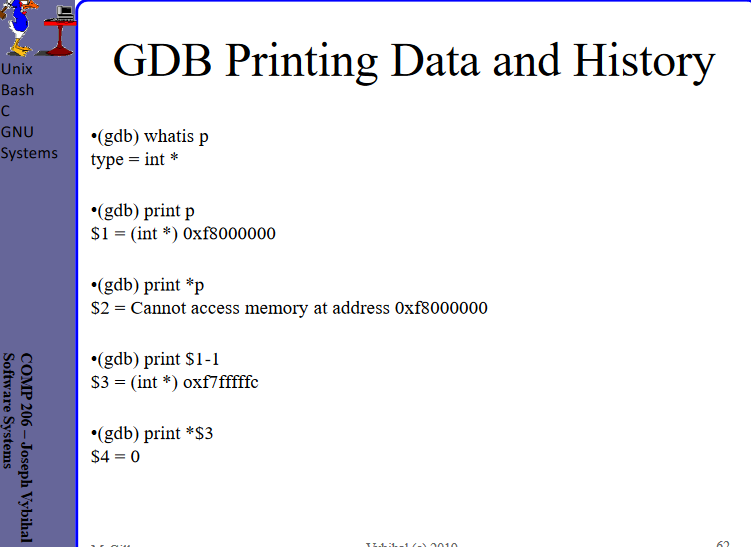
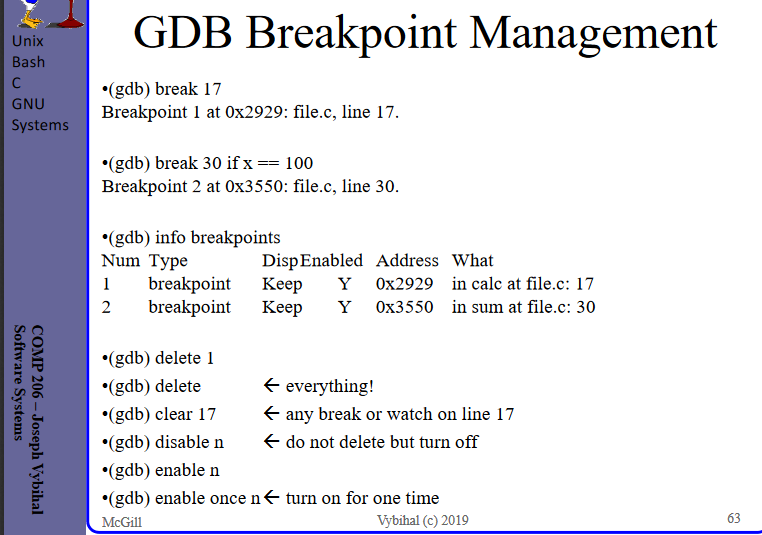
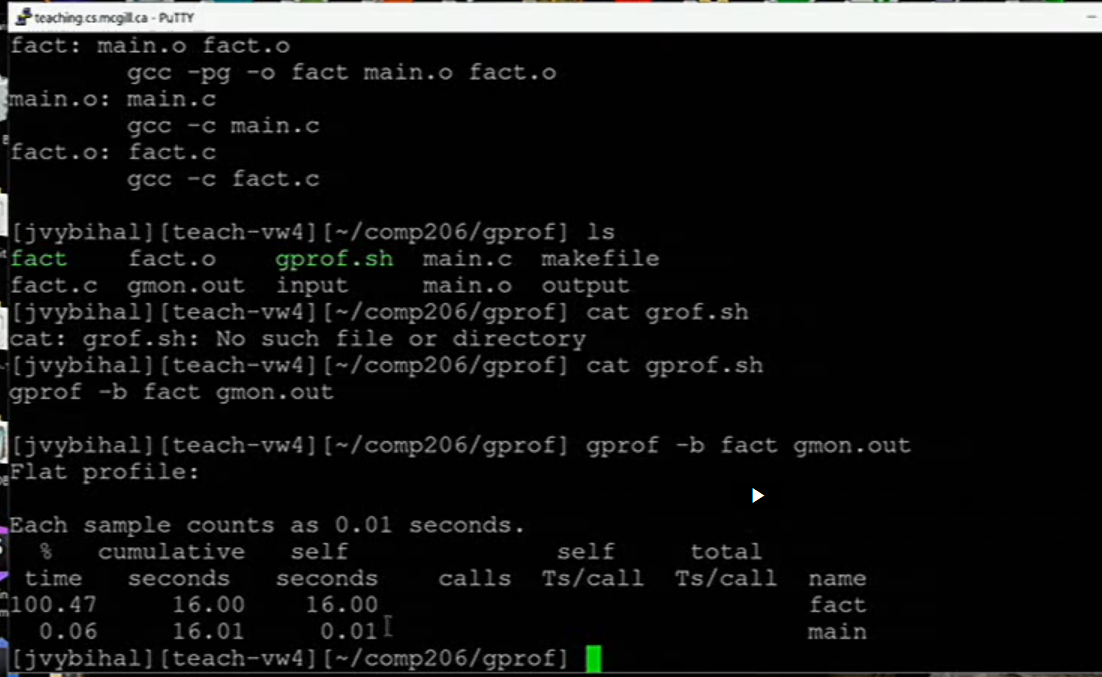
COMP206 Notes:

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. **Modular programming:** a program composed from multiple objects:
15. 
16. **#include directive has two formats:** #include<library> and #include “path/textfile”
17. 
18. **The gcc –c command**: compiles each source file separately saving each .c file as a compiled .o file.
19. **The gcc command by itself:** merges all the .o files into an a.out executable file
20. **Each .o file’s global variables are, by default:** local to only the .o file it was defined and compiled within
21. **In contrast, all function names:** are public but “hidden”
22. **Using the extern command a program can**: access a semi-private variable
23. 
24. 
25. 
26. **.h files**: allow developers to share semi-private information with other developers without verbal communication. (or choosing not to share)
27. 
28. 
29. **#define TERM EXPRESSION:** the EXPRESSION will be inserted into the source code everywhere it finds TERM
30. **#ifdef or #ifndef TERM:** Check if the user defined something
31. 
32. **GNU:** Open source project to create a Unix-like but free environment, contains Unix-like operating system: HURD kernel and many development tools
33. **The GNU make tool:** helps us compile complicated projects and do other software engineering activities, like backups, etc.
34. **Make Tool is composed of:** make program and makefile text file
35. **Makefile:** file in which developers define their software engineering rules
36. **Make program:** uses that text file
37. 
38. **Method 1:** $ make, which assumes a makefile exists in the same directory and executes the first line
39. **Method 2:** $ make backup, which invokes the make program and specifically targets a portion of the makefile.
40. **Big Oh Advantages and Problems:** true speed independent from hardware, but long to do on large pieces of code
41. **Clock:** can be automated, but get better CPU, program runs faster
42. **Profilers:** record data as the application executes
43. 
44. 
45. 
46. **Call graph:**



1. 
2. 
3. 
4. **GDB is:** a debugger
5. **Steps for GDB:** first gcc -g, then gbd (executable)
6. **To list lines, use:** “list” command
7. **To analyse core-dump:** gdp -g, then gdb a.out core-dump
8. **To analyze a certain line:** use “break” and “run”
9. 
10. 
11. 
12. **Where command to:** produce a backtrack
13. 
14. Types of errors: Syntax errors, runtime errors, doesn’t crash but wrong answer
15. **Syntax error:** compiler finds error
16. **Runtime error:** compiler doesn’t find but program crashes
17. You can step into a function like scanf and analyze lines of code. If you just want to move on to the next line and skip the scanf, you can go “next” in GDB.
18. 