# CS 354 - Machine Organization & Programming Tuesday April 23 , and Thursday April 25th, 2024

**Homework hw7:** DUE on or before Monday Apr 21st **Homework hw8:** DUE on Monday Apr 29th **Homework hw9:** DUE on Wednesday May 1st

**Project p6:** Due on last day of classes, May 3rd. **Please complete p6 by Friday of this week as labs are very busy last week of classes**.

If you do plan on getting help during last week of classes, be sure to bring your own laptop in case there is no workstation available.

**Last Week**

|  |  |
| --- | --- |
| Pointers | Transferring Control via Exception Table |
| Function Pointers | Exceptions/System Calls in IA-32 & Linux |
| Buffer Overflow & Stack Smashing | Processes and Context |
| Flow of Execution | User/Kernel Modes |
| Exceptional Events | Context Switch |
| Kinds of Exceptions | Context Switch Example |

**This Week**

|  |  |
| --- | --- |
| Meet Signals  Three Phases of Signaling Processes IDs and Groups Sending Signals  Receiving Signals | Issues with Multiple Signals Forward Declaration Multifile Coding  Multifile Compilation Makefiles |
| **Next Week**: Linking and Symbols B&O 7.1 Compiler Drivers   * 1. Static Linking   2. Object Files   3. Relocatable Object Files   4. Symbols and Symbols Tables   5. Symbol Resolution   6. Relocation | |

# Meet Signals

A paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A white paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A screenshot of a paper with text

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

*A paper with writing on it

Description automatically generated*

### gcc Compiler Driver

preprocessor compiler assembler linker

### Object Files

relocatable object file (ROF)

executable object file (EOF)

shared object file (SOF)

### Compiling All at Once

gcc align.c heapAlloc.c -o align

### Compiling Separately

gcc -c align.c

gcc -c heapAlloc.c

gcc align.o heapAlloc.o -o align

##  Compiling separately is

*A white paper with writing on it

Description automatically generated*