

Name \_\_\_\_\_ G# \_\_\_\_\_

Group Member Name: \_\_\_\_\_

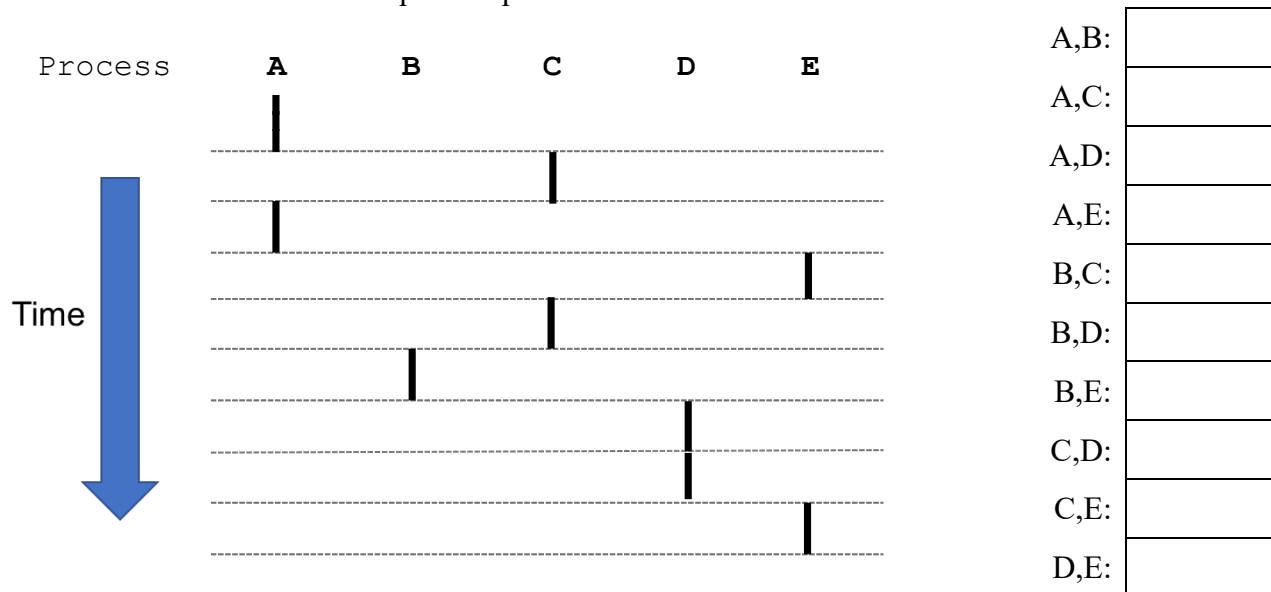
Group Member Name: \_\_\_\_\_

**Today's Goals:** We want to get familiar with basic exceptional control flows and looking at processes.**Work in groups of 2-3 students.** Every group will turn in what they've got to Blackboard.

**Grading is based on participation.** Get as much done as you can. You will also be given feedback in the form of a 'score' (1-3) and possibly some comments. This doesn't affect your grade – it is solely for feedback. A score of 3 means everything looks great. A score of two indicates some minor problems. And a score of one indicates that there were some major issues. If you get a 1, don't panic - go see your prof or a GTA to get more extensive feedback.

## Processes and concurrency

1. Assuming single-core machine, identify each pair of processes as concurrent or sequential processes. Put 'C' for concurrent and 'S' for sequential processes.



2. What is the mechanism that creates an illusion that each process has an exclusive use of the CPU?
3. How are processes created, loaded, and reaped? Explain using `fork()`, `exec()`, and `wait()`.
4. What is reaping and why is it needed?

## Process creation and termination

Consider the following C program.

```
fun1() {  
    printf("A");  
    if (fork()){  
        printf("B");  
        if (!fork()) printf("C");  
        printf("D");  
        exit(0);  
    }  
    printf("E");  
    fork();  
    printf("F");  
}
```

1. Draw the process graph of **fun1()**. Use the text area below or upload additional files for your graph.

2. In **fun1()**, how many times will each string ("A", "B", "C", "D", "E", "F") be output?

A: \_\_\_\_\_

B: \_\_\_\_\_

C: \_\_\_\_\_

D: \_\_\_\_\_

E: \_\_\_\_\_

F: \_\_\_\_\_

3. In **fun1()**, which of the following outputs are legal?

- |             |       |         |
|-------------|-------|---------|
| a) AEFFBDCD | Legal | Illegal |
| b) AEBDFCDF | Legal | Illegal |
| c) ABFCDEDF | Legal | Illegal |
| d) ABDDECFE | Legal | Illegal |
| e) ABCEDFFD | Legal | Illegal |

4. Consider the following C program.

```
int main() {  
    if (fork()) {  
        printf("A");  
        waitpid(-1, null, 0);  
    }  
    else {  
        printf("B");  
        exit(0);  
    }  
    printf("C");  
    return 0;  
}
```

(a) Draw the process graph. Use the text area below or upload additional files for your graph.

(b) What are all of the possible output sequences of the program?

(c) Redraw the process graph assuming the `waitpid()` call is removed. Use the text area below or upload additional files for your graph.

(d) If the `waitpid()` call is removed, what are all of the possible outputs?