

EECE.4810/EECE.5730: Operating Systems

Spring 2018

Lecture 3: Key Questions

January 29, 2018

1. **Example:** Including the initial parent process, how many processes does the program below create? Draw a process tree to support your answer.

```
int main() {  
    for (int i = 0; i < 4; i++)  
        fork();  
  
    return 0;  
}
```

2. **Example:** What does the program below print?

```
int nums[5] = {0,1,2,3,4};

int main() {
    int i;
    pid_t pid;

    pid = fork();

    if (pid == 0) {
        for (i = 0; i < 5; i++) {
            nums[i] *= -i;
            printf("CHILD: %d\n", nums[i]);
        }
    }
    else if (pid > 0) {
        wait(NULL);
        for (i = 0; i < 5; i++)
            printf("PARENT: %d\n", nums[i]);
    }
}
```

3. Describe the purpose of the various exec system calls.

4. Review the operation of the example code below (which is slightly modified from last Wednesday's lecture)

```
int main() {
    pid_t pid;
    pid = fork();           // Create a child process

    if (pid < 0) {          // Error occurred
        fprintf(stderr, "Fork failed");
        return 1;
    }
    else if (pid == 0) {    // Child process
        printf("Child: listing of current directory\n\n");
        execlp("/bin/ls", "ls", NULL);
    }
    else {                 // Parent process—wait for child to complete
        printf("Parent: waits for child to complete\n\n");
        wait(NULL);
        printf("Child complete\n\n");
    }
    return 0;
}
```