

# EECE.4810/EECE.5730: Operating Systems

Spring 2017

## Lecture 3: Key Questions

January 25, 2017

1. (Review) Describe the operation of this basic program, which ultimately represents two separate processes.

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        execlp("/bin/ls", "ls", NULL);
    }
    else { /* parent process */
        /* parent will wait for the child to complete */
        wait(NULL);
        printf("Child Complete");
    }

    return 0;
}
```

2. Describe how processes are terminated.

3. What are the two models of interprocess communication?

4. Describe the basics of the producer-consumer problem.

5. Describe the following pseudo-code, which represents a bounded-buffer implementation of a producer-consumer setup using shared memory IPC.

```
// Basic setup
#define BUFFER_SIZE 10
typedef struct {
    . . .
} item;
item buffer[BUFFER_SIZE];
int in = 0;
int out = 0;

// Producer
item next_produced;
while (true) {
    /* produce an item in next produced */
    while (((in + 1) % BUFFER_SIZE) == out)
        ; /* do nothing */
    buffer[in] = next_produced;
    in = (in + 1) % BUFFER_SIZE;
}

// Consumer
item next_consumed;
while (true) {
    while (in == out)
        ; /* do nothing */
    next_consumed = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    /* consume the item in next consumed */
}
```

6. Describe the basics of interprocess communication through message passing.

7. How do processes communicate using direct communication?

8. How do processes communicate using indirect communication?