

EECE.4810/EECE.5730: Operating Systems

Spring 2017

Lecture 4: Key Questions

January 30, 2017

1. (Review) What are the two models of interprocess communication? What are the benefits of each?

2. Describe the basics of shared memory, using the POSIX shared memory producer/consumer example programs below (producer on this page; consumer on next):

```
int main()
{
    /* the size (in bytes) of shared memory object */
    const int SIZE = 4096;
    /* name of the shared memory object */
    const char *name = "OS";
    /* strings written to shared memory */
    const char *message_0 = "Hello";
    const char *message_1 = "World!";

    /* shared memory file descriptor */
    int shm_fd;
    /* pointer to shared memory object */
    void *ptr;

    /* create the shared memory object */
    shm_fd = shm_open(name, O_CREAT | O_RDWR, 0666);

    /* configure the size of the shared memory object */
    ftruncate(shm_fd, SIZE);

    /* memory map the shared memory object */
    ptr = mmap(0, SIZE, PROT_WRITE, MAP_SHARED, shm_fd, 0);

    /* write to the shared memory object */
    sprintf(ptr, "%s", message_0);
    ptr += strlen(message_0);
    sprintf(ptr, "%s", message_1);
    ptr += strlen(message_1);

    return 0;
}
```

2 (continued) POSIX shared memory consumer:

```
int main()
{
/* the size (in bytes) of shared memory object */
const int SIZE = 4096;
/* name of the shared memory object */
const char *name = "OS";
/* shared memory file descriptor */
int shm_fd;
/* pointer to shared memory object */
void *ptr;

/* open the shared memory object */
shm_fd = shm_open(name, O_RDONLY, 0666);

/* memory map the shared memory object */
ptr = mmap(0, SIZE, PROT_READ, MAP_SHARED, shm_fd, 0);

/* read from the shared memory object */
printf("%s", (char *)ptr);

/* remove the shared memory object */
shm_unlink(name);

return 0;
}
```

3. Describe message passing through direct communication.

4. Describe message passing through indirect communication.

5. Describe the three forms of IPC used in client-server systems:
a. Sockets

b. Remote procedure calls

c. Pipes