

EECS 678 – Operating Systems – Fall 2020  
Quiz – 3

Name: Prasad Kulkarni

Lab Time: (a) M 12:00pm, (b) M 4:00pm, (c) W 9:00am, (d) W 4:00pm, (e) F 12:00pm

1. True or **False** (1 point): On Linux, the parent process always waits for its child process to finish.
2. **True** or False (1 point): The *shared memory* IPC model will typically allow faster communication than *message passing*.
3. True or **False** (1 point): In the *shared memory* IPC model, reads to shared memory are typically blocking.
4. True or **False** (1 point): Any two processes running on the same OS can use the (anonymous) *pipe* IPC mechanism to communicate with each other.
5. Fill-in the blank (1 point): The FIFO / named pipe IPC mechanism limits a process to be a *reader* or a *writer*, but not both at once.
6. Select the correct answer (1 point): Which IPC mechanism allows sent messages to be received out-of-order?  
(a) Pipe (b) Fifo **(c) Message queue**
7. Fill-in the blanks (2 points): The two models for interprocess communication (IPC) are: shared memory and message passing.

Please turn page over

8. Answer (3 points): Use the pipe IPC mechanism to synchronize the parent and child processes below to guaranty the following output:

From Child process  
From Parent process

```
int main()
{
    char *s, buf[1024];
    int fds[2];
    char *s = "Pipe program for process synchronization\n";

    /* create a pipe */
    1pt - pipe(fds);
    if (fork() == 0) {

        printf("From Child process\n");
        1pt - write(fds[1], s, strlen(s));
    } else {

        1pt - read(fds[0], buf, strlen(s));
        printf("From Parent process\n");

    }
}
```