1. In the context of systems programming, what is a signal?

2. There are three possible dispositions of signals. Name them.

3. How many standard signals are there?

4. What are the two ways that can be used to change the disposition of a signal?
5. If you want to reset the disposition of a signal to its default, there is a constant that can be used in place of a handler function. What is it?

6. How does one send a signal to a process?

7. When are signals handled? Does this pose any problems for any other functions?

8. The man page for `sigaction` claims that `sa_handler` and `sa_sigaction` should never both be defined. Why is this?

9. How does one define a signal set to be used for functions such as `sigpending`?
Part 2 - IPC

Answer 6 of the questions below. The remaining 3 can be answered for extra credit. Please clearly mark which are which, otherwise it will be assumed 1-6 count and 7-9 are the extra credit.

1. How do pipes and FIFOs differ?

2. Provide the pseudocode for a select loop.

3. There are three POSIX IPC mechanisms, as defined in the POSIX.1b realtime extensions. Please list them and provide a succinct summary of their purpose.

4. Describe how to use shared memory.

5. Describe how to remove a shared memory segment (or object, if you prefer).
6. What issues must be addressed when dealing with IP addresses and port numbers when passing these values across a network?

7. What data structure is used to store an IPv4 socket address?

8. When select returns, it provides the number of file handles which are ready for I/O. How does one determine which file handles in the set are in a ready state?

9. When opening a FIFO, code blocks until both ends are open. How is this avoided?

**Part 3 - Synchronization**

1. What are the two types of semaphores that can be used?
2. There are 4 main operations on unnamed semaphores. Please list them, and describe their purpose.

3. Consider the project. Please describe the process of checking and updating the bitmap.

4. Describe how the two types of semaphores are used, and how they differ.

5. What is an *async-signal-safe* function?

**Part 4 - Threads**

_In class, we briefly discussed the concept of threading. The following questions deal with this concept. Answer 3 of these. Same rules as previous sections._

1. We discussed the 2 ways in which you can avoid zombie threads. Describe one of them.
2. What is the prefix that is used for all of the thread functions we discussed?

3. What is the advantage of threading over creating multiple processes?

4. How do threads differ from processes, in Linux?