

# 11/5/15 Pre-Class Work

\* Required

**Harvard email address: \***

**Please indicate which course you are taking. \***

- CS61 (College)
- CSCIE-61 (Extension)

**Extension students only:**

Would you prefer that web-conferences were:

- walk throughs of the exercises
- small group work
- a combination of walk through and small group work
- Other:

**Extension students only:**

If you chose "combination" what would be the ideal way to split up the session?

## Practicum

The practicum is a new section of pre-class work designed to make sure you are developing the practical skills we think you should be developing.

I will try to include one question on new material and one or two questions on things that should be old hat by now.

**The signal system call we introduced in the video and used in class is actually a simplification of the sigaction system call.**

Read the man page for sigaction. Write whatever lines of code are necessary to express the invocation of `signal(SIGINT, int_handler)` in terms of `sigaction`.

```

struct sigaction act;
sigemptyset (&act.sa_mask)
act.sa_flags = 0;
act.sa_handler = int_handler;
sigaction(SIGINT, &act, NULL);

```

The git command "git add remote handout some\_repo" does what: \*

- Pulls an uptodate version of your origin repository.
- Commits your current changes locally.
- Adds a local name for "some\_repo."
- Changes the contents of your current repository to the contents of some\_repo.

Using git, it is possible to pull changes from more than one repository. \*

- True
- False

## Select

Which explanation below best describes the "check and wait problem?" \*

- The same process cannot check for an event and wait for that event.
- There is no way to communicate events between different processes.
- We need a way to atomically check for an event and wait if the event has not happened.
- We need to block signals so that you don't get interrupted while waiting for an event.

In the video, the signal pipe is read from and written to by: \*

- The parent and the parent
- The parent and ping
- The parent and pong
- ping and pong

What functionality does pselect offer that select does not? \*

- The ability to wait a limited amount of time for events on file descriptors.
- The ability wait on both read and write file descriptors.
- The ability to wait on a child with timeout.
- The ability to block or unblock a set of signals while waiting for the event.

## Synchronization

**Which of the following characteristics will prevent deadlock from occurring? \***

- All threads acquire resources in the same order.
- Share resources via processes instead of threads.
- Do not use blocking synchronization primitives.
- Keep critical sections short.

**If you have a race condition in your code, you will always observe errant behavior. \***

- True
- False

**A process can have multiple threads of control. \***

- True
- False

**Which of the following are things that threads share but processes do not? \***

- Global variables
- Heap variable
- File descriptors
- Process IDs
- Thread IDs

**In pthreads, if you create a condition variable, it automatically creates a mutex for it. \***

- True
- False

Submit

*Never submit passwords through Google Forms.*

Powered by

This form was created inside of Google Apps for Harvard.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)