10/29/15 Pre-Class Work

* Required

Harvard email address:

Please indicate which course you are taking:

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

Signals

All signals can be blocked

- True
- False

All signals can be ignored

- True
- False

We use the signal system call to send signals to another process

- True
- False

Code sample A

```c
sigset_t set;

sigemptyset(&set);
sigaddset(&set, SIGHUP);
sigprocmask(SIG_BLOCK, &set, NULL);
sigdelset(&set, SIGHUP);
sigaddset(&set, SIGCONT);
sigprocmask(SIG_BLOCK, &set, NULL);
```

The sequence of calls in code sample A will leave the signals in what state?
No signals blocked
SIGHUP blocked
SIGCONT blocked
Both SIGHUP and SIGCONT blocked

Code sample B

```c
sigset_t set;

sigemptyset(&set);
sigaddset(&set, SIGHUP);

sigprocmask(SIG_BLOCK, &set, NULL);
sigdelset(&set, SIGHUP);
sigaddset(&set, SIGCONT);
sigprocmask(SIG_SETMASK, &set, NULL);
```

The sequence of calls in code sample B will leave the signals in what state? *

- No signals blocked
- SIGHUP blocked
- SIGCONT blocked
- Both SIGHUP and SIGCONT blocked

Which of the following are probably fine things to do in a signal handler. *

- Sleep
- Call malloc
- Call printf
- Examine a variable
- Unblock signals
- Write a global variable
- Install a new signal handler

Blocking and Polling

Polling can be a superior solution to blocking when: *

Check all that apply

- The process needing to block has other work it needs to get done.
- The condition is easy to check
- The event for which we want to wait won't happen for a long time
- Blocking will take more time than the event for which we’re waiting

https://docs.google.com/forms/d/1zHj5ZpMx6LZjENnuYHhq1cDc6p90fXRe-iQoL_g1tgRw/viewform
Achieving 100% utilization is always a good thing *

☐ True
☒ False

waitpid can be used in either blocking or polling modes *

☒ True
☐ False

waitpid can return EINTR *

☐ True
☒ False

returns -1 and can set errno to EINTR

Never submit passwords through Google Forms.