Advanced Programming
COMS 3157

April 18, 2025

Patrick Shen

pts2125@columbia.edu

---

1. Q1? (2 marks)
   (a) What is a signal?
   (b) What is a signal handler?
2. Do the signal() and sigaction() methods pause the flow of code? (1 mark)
3. How would each of these signals be triggered? (4 marks)
   (a) SIGFPE
   (b) SIGINT
   (c) SIGTSTP
   (d) SIGCONT
4. Which two signals cannot be handled? (2 marks)
5. Briefly explain each argument in for sigaction(int signum, const struct sigaction *act, struct sigaction *oldact) (3 marks)
   (a) int signum
   (b) struct *act
6. What do each field for in the sigaction struct? (4 marks)
   (a) void (*sa_handler)(int);
   (b) void (*sa_sigaction)(int, siginfo_t *, void *);
   (c) sigset_t sa_mask;
   (d) int sa_flags;

```c
struct sigaction {
    void     (*sa_handler)(int);
    void     (*sa_sigaction)(int, siginfo_t *, void *);
    sigset_t   sa_mask;
    int        sa_flags;
    void     (*sa_restorer)(void); // obsolete, ignore
};
```

```

```

Listing 1: sigaction struct

7. Briefly explain what each function does for sa_mask in the sigaction struct (3 marks)
   (a) int sigemptyset(sigset_t *set)
   (b) int sigaddset(sigset_t *set, int signum)
   (c) int sigfillset(sigset_t *set)

8. What does the function call memset() do here? (1 mark)

```
    struct sigaction act;

    memset (&act, '\0', sizeof(act));

```

Listing 2: memset()

9. What does act = {0} do here? (1 mark)

```
    struct sigaction act;

    act = {0};

```

Listing 3: act

10. Suppose a SIGTERM signal comes in. What is the output? (1 mark)

```
    static void hd1 (int sig, siginfo_t *siginfo, void *context)
    {
        printf("SIGTERM receieved.");
    }

    ....

    struct sigaction act;

    memset (&act, '\0', sizeof(act));

    act.sa_sigaction = &hd1;
    act.sa_flags = SA_SIGINFO;

    if (sigaction(SIGTERM, &act, NULL) < 0)
    {
        perror("sigaction");
        return 1;
    }

```

Listing 4: simple example

11. What do the following keywords in C do? (2 marks)

    (a) volatile

    (b) sig_atomic_t

```c
volatile sig_atomic_t signal_val = 0;
```

Listing 5: keywords

12. What does the raise(int iSig) function do? (1 mark)

13. What does the kill(pid_t iPid, int iSig) function do? (1 mark)

14. What is the output for each of these commands? The code is stored in a executable named "sleep". (2 marks)

    (a) ./sleep 2 (Ctrl + C is not sent)

    (b) ./sleep 5 (Ctrl + C is sent 4 seconds in)

```c
    void catch_signal(int sig) {
        got_signal = 1;
    }

    int main(int argc, char *argv[]) {
        if (argc != 2) {
            fprintf(stderr, "Usage: %s <seconds>\n", argv[0]);
            return EXIT_FAILURE;
        }

        int max_snooze_secs = atoi(argv[1]);
        if (max_snooze_secs <= 0) {
            fprintf(stderr,
            "Error: Invalid number of seconds '%s' for max snooze
   time.\n",
            argv[1]);
            return EXIT_FAILURE;
        }

        struct sigaction action = {0};
        action.sa_handler = catch_signal;
        action.sa_flags = SA_RESTART;
        if (sigaction(SIGINT, &action, NULL) == -1) {
            perror("sigaction");
            return EXIT_FAILURE;
        }

        int count = 0;
        while (!got_signal && count < max_snooze_secs) {
            sleep(1);
```

```
30              count++;
31          }
32          printf("Slept for %d of the %d seconds allowed.\n",
33          count, max_snooze_secs);
34
35          return EXIT_SUCCESS;
36      }
37
38
```

Listing 8: kill() example

15. Answer the following questions about the alarm(int time) function. (2 marks)
    (a) What does the alarm(int time) function do?
    (b) What happens if the time argument is set to 0?