CS 551 Systems Programming, Summer 2024

Homework Assignment 4

Out: 8/1/2024 Thur.

Due: 8/10/2024 Sat. 23:59:59

Q&A (100 points)

1. (20 points) Message queues are typically used for exchanging information in a server/client application framework. If a malicious process reads a message from a message queue that is being used by a server and several clients, explain

(1) What will happen to the server/client?

(2) What criteria need to be met in order for the malicious process to read the message queue? Explain for POSIX message queues.

2. (20 points) Compare FIFOs with POSIX message queues (i.e., list the similarities and differences between the two IPC facilities).

3. (20 points) Compare POSIX binary semaphores (a semaphore whose value is restricted to be 0 or 1) with Pthread mutexes.

4. (20 points) Read the following code, and answer the questions that follow. (Note that for simplicity, the necessary return value checking is omitted in the code.)

```
static void notifySetup(mqd_t *mqdp);

static void threadFunc(union sigval sv)
{
    ssize_t numRead;
    mqd_t *mqdp;
    void *buffer;
    struct mq_attr attr;

    mqdp = sv.sival_ptr;
    mq_getattr(*mqdp, &attr);
    buffer = malloc(attr.mq_msgsize);
    notifySetup(mqdp);

    while ((numRead = mq_receive(*mqdp, buffer, attr.mq_msgsize, NULL)) >= 0)
        printf("Read %ld bytes\n", (long) numRead);

    free(buffer);
}

static void notifySetup(mqd_t *mqdp)
{
    struct sigevent sev;

    sev.sigev_notify = SIGEV_THREAD;
    sev.sigev_notify_function = threadFunc;
27     sev.sigev_notify_attributes = NULL;
28     sev.sigev_value.sival_ptr = mqdp;
29     mq_notify(*mqdp, &sev);
30 }
31
32 int main(int argc, char *argv[])
33 {
34     mqd_t mqd;
35     mqd = mq_open(argv[1], O_RDONLY | O_NONBLOCK);
36
37     notifySetup(&mqd);
38     pause();                    /* Wait for notifications via thread function */
39 }
```

(1) Describe what the program does.

(2) What is the purpose of line 13 (i.e., "`notifySetup(mqdp)`")? Why do we need to call it again given that it is already called in the `main` function (line 37)?

(3) Can we make the `buffer` in `threadFunc` (line 7) a global variable, and allocate its memory just once in the main program? Explain your answer.

5. (20 points) Read the following code (a daytime server), and answer the questions that follow. (Note that for simplicity, the necessary return value checking is omitted in the code.)

```
1 int main(int argc, char **argv)
2 {
3     int listenfd, connfd;
4     socklen_t len;
5     struct sockaddr_in servaddr, cliaddr;
6     char    buff[1024];
7     time_t  ticks;
8
9     listenfd = socket(AF_INET, SOCK_STREAM, 0);
10
11     bzero(&servaddr, sizeof(servaddr));
12     servaddr.sin_family = AF_INET;
13     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
14     servaddr.sin_port = htons(13);  /* daytime server */
15
16     bind(listenfd, (SA *) &servaddr, sizeof(servaddr));
17
18     listen(listenfd, 1000);
19
20     for ( ; ; ) {
21         len = sizeof(cliaddr);
22         connfd = accept(listenfd, (SA *) &cliaddr, &len);
23         ticks = time(NULL);
24         snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));
25         write(connfd, buff, strlen(buff));
26         close(connfd);
27     }
28 }
```

(1) If we remove `listen()` (line 18), what will happen?

(2) If we remove `bind()` (line 16) (and keep `listen()`), what will happen?

Submission instructions

- Type your answers using whatever text editor you like, remember to include the index number of each question.
- Export the file to PDF format.
- Name the PDF file based on your BU email ID. For example, if your BU email is "abc@binghamton.edu", then the PDF file should be named as "hw4_abc.pdf".
- Submit the PDF file to Brightspace website before the deadline.

Grading guidelines:

(1) If the submitted PDF file is not named as specified above (so that it causes problems for TA's automated grading scripts), 10 points off.

(2) Lastly but not the least, stick to the collaboration policy stated in the syllabus: you may discuss with your fellow students, but code should absolutely be kept private.