UNX511NZA Midterm Answers

Name: Aryan Khurana
Student ID: 145282216
Email: akhurana22@myseneca.ca

Question 1: How many binaries are created if "make all" is executed from the command line?
Answer) 3 Binaries, processMon, memMonitor, fileMon

Question 2: What happens if the following is executed from the command line: "pkill -2 processMon"
Answer) The signal handler (static void sigHandler(int sig)) is called and it prints the message "processMonitor: SIGINT received". The isRunning variable is also set to false, causing the processMonitor() loop to stop.

Question 3: What function in the kernel is called when open is called here?
Answer) sys_open() is called

Question 4: How many processes are running at this point?
Answer) 3, one parent and two child processes.

Question 5: What function in the kernel is called when read is called here?
Answer) sys_read() is called

Question 6: Why is it an error if the program continues beyond execlp()?
Answer) execlp() replaces the current process image with a new process image. If it is successful, the code following it will not execute. If there is an error the program continues executing the next lines.

Question 7: What function in the kernel is called when ioctl is called here?
Answer) sys_ioctl() is called

Question 8: What function in the child processes is called when they receive a SIGINT?
Answer) The signal handler (static void sigHandler(int sig)) is called.

Question 9: What are we waiting on here?
Answer) We are waiting for any child process to terminate. The wait() function does not call the parent process until one of its child processes exits. Once the child process exits, the function retrieves the child's exit status.

Question 10: What is processMon attempting to do with the log file?
Answer) the processMon program attempts to open a log file using the open function with specific flags and permissions. The flags O_RDWR, O_CREAT, and O_APPEND are used to specify that the file should be opened for reading and writing (O_RDWR), created if it does not already exist (O_CREAT), and any writes to the file should append to the end rather than overwriting existing content (O_APPEND). The mode_t filePerms is set to S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH, which grants read and write permissions to the file owner (user), read and write permissions to the group, and read and write permissions to others (everyone).

Question 11: What is processMon attempting to do with the log file now?
Answer) processMon is trying to set the O_EXCL flag on the log file descriptor, which means that one process will be given exclusive access to this file and that the log file should only be opened if it does not already exist. If the log file already exists, this call will fail, indicating an error condition. The goal is to ensure that only one instance of the log file is written to at a time.

Question 12: Why are we invoking execlp() again here?
Answer) This block of code will restart the child process with a new PID after it has been terminated. When the original child process (which runs either fileMon or memMonitor) exits, the processMon program detects this termination and attempts to create a new child process to replace it.

Question 13: This appears to be reading a file under /dev. What is it really doing?
Answer) This line reads data from the /dev/keygen file, which is a character device. The read function used in user space needs to be linked to the kernel function in the device driver.

Question 14: What happens if memMonitor receives a SIGKILL?
Answer) If memMonitor receives a SIGKILL, the operating system will immediately terminate the process. Since SIGKILL cannot be caught or ignored, it is not explicitly handled in the function.

Question 15: What happens if memMonitor receives a ctrl-C?
Answer) The sigHandler(int sig) function will print the message "memMonitor: SIGINT received" and set the isRunningvariable to false. This action will stop the while loop, effectively terminating the program.

Question 16: Describe these open flags
Answer) O_RDWR opens the file for both reading and writing, O_CREAT creates the file if it does not already exist and with O_APPEND, if the file exists, new data will be added to the end of the file instead of overwriting the existing content.

Question 17: Describe these file permissions
Answer) S_IRUSR grants read permissions to the file owner, S_IWUSR grants write permissions to the file owner, S_IRGRP grants read permissions to the group associated with the file, S_IWGRP grants write permissions to the group associated with the file, S_IROTH grants read permissions to others (users not in the file's owner or group), S_IWOTH grants write permissions to others (users not in the file's owner or group).

Question 18: Where is key1 coming from?
Answer) key1 holds the value of the first command-line argument, which is copied from argv[1] using the strcpy function: strcpy(key1, argv[1], MAXBUF).

Question 19: Where is key2 coming from?
Answer) key2 is poppulated with data by reading from the /dev/keygen device file

Question 20: Why are they being compared?
Answer) The program checks if key1, provided by the user, matches key2, which is obtained from the device. If they do not match, the program indicates that the key is invalid. This acts as a security measure to ensure that only authorized users can access the program's features.

Question 21: What do the following two lines effectively do to the log file?

Answer) The line fd = open(logFile, openFlags, filePerms); opens the log file (midterm.log) for reading and writing. It creates the file if it doesn't already exist and appends data to it if it does. The close(fd); function then closes the file descriptor, releasing the resources associated with it.

Question 22: Is keyFile (/dev/keygen) a file that we can edit with vim?
Answer) No you cannot edit this file using vim. It is a special device file representing a character device in the Linux filesystem, specifically designed for interacting with kernel-level features or hardware.

Question 23: When we execute ls -l on /dev/keygen, we see the following:
Question 23 (continued): crw-rw-rw- 1 root root 238, 0 Feb 11 11:30 /dev/keygen
Question 23 (continued): What does the c mean in crw-rw-rw-?
Answer) c indicates that this is a character device driver. It looks like a file but it is not.

Question 24: What does the number 238 represent?
Answer) 238 is the major device number assigned to this particular character device driver.