Lab 12 (CS): Input/Output

Shokhista Ergasheva, Muwaffaq Imam, Artem Kruglov,
Nikita Lozhnikov, Giancarlo Succi, Xavier Vasquez
Herman Tarasau, Firas Jolha

Innopolis University
Course of Operating Systems

Week 12 – Lab

Exercise 1 (1/2)

Background:

- A peripheral device is controlled by writing and reading its registers. Often, a device has multiple registers that can be accessed at consecutive addresses either in the memory address space or in the I/O address space. Each device connected to the I/O bus has a set of I/O addresses, called I/O ports. I/O ports can be mapped to physical memory addresses so that the processor can communicate with the device through instructions that work directly with the memory.

Exercise 1 (2/2)

Description + Constraints:

- Get serial ports from */proc/ioports* using *sudo* permission and save the output to `ex1.txt`.
- What are `dma1`, `pic1` and `timer1`? What do they represent? Write your answers to `ex1.txt`.
- Submit `ex1.txt`.

Note: If you are getting zero-valued addresses when accessing the file, then use `sudo` permission.

Exercise 2 (1/2)

Description:

- Write a C program `ex2.c` that gets keyboard events directly from the keyboard device and prints it to stdout.

Note: Try exploring
`/dev/input/by-path/platform-i8042-serio-0-event-kbd`

Exercise 2 (2/2)

Constraints:

- You have to use the file `/dev/input/by-path/platform-i8042-serio-0-event-kbd` for capturing keyboard events.
- You should use the `input_event` structure from `linux/input.h`[1]
- Only `PRESSED` and `RELEASED` events should be handled.
- Print the output events in format: **PRESSED 0x0023 (35)**
  - Where **PRESSED** - type of event, **0x0023** and **(35)** are hex and decimal representation of event code respectively.
- The progam should be executed using sudo permission.
- Print and save the output to `ex2.txt`
- Save the code in `ex2.c`
- Submit `ex2.txt` and `ex2.c`

[1] `https://www.kernel.org/doc/Documentation/input/input.txt`

Exercise 3 (1/2)

Description:

- Modify previous program to output only shortcuts either: P+E → "I passed the Exam!", C+A+P → "Get some cappuccino!" and one custom shortcut of your choice. Save code in `ex3.c` and sample output in `ex3.txt`
- **Notes: (added to clear up some ambiguity)**
  - The user should press all keys of the shortcut to print the message. For instance, to print the message "I passed the Exam!", the user presses P while holding down E. You must handle the case when the user is holding down other keys with the keys of the shortcut. For instance, the user pressed P+E+A should not print the message. There is no order in pressing the keys of the shortcut. For instance, the user can press P then E or E then P.
  - When the keys of the shortcut in REPEATED state, the program should repeatedly print the message until one of the keys at least are released. For instance, the user is holding down P and E, the message should be printed until the user releases E or P or both.

Exercise 3 (2/2)

Constraints:

- The program should print only on specified shortcuts.
- The shortcut P + E, should print ”I passed the Exam!” to stdout.
- The shortcut C + A + P, should print “Get some cappuccino!” to stdout.
- Create the shortcut of your choice with custom message. The number of keys should be at least 2 and no more than 6.
- Print the available shortcuts at start of the program.
- Print and save the output to `ex3.txt`.
- Save the code in `ex3.c`.
- Submit `ex3.txt` and `ex3.c`.

End of lab 12 (CS)