Name: Truong-Giang Pham
Ccid: truonggi

Objectives: state the project objectives and value from your point of view (which may be different from the one mentioned above) –

The objective of this assignment was to get familiar with interprocess communication with named pipes. Not only that, the use of I/O multiplexing was an essential part of this assignment. With poll(), I was able to learn how to manage the communication between so many fifos at once. Although being a small part of the specification, I was able to get exposed to signal handling in C, and that reinforced my learning about signal processing of the operating system. I would love to try to implement "delay" using threads, but time was not in my favor to try this approach.

Design Overview: highlight in point-form the important features of your design

- **Assumption**:Run the master switch first, then each pswi switch in increasing order
- **Assumption:** create all the fifos first using the command line, or type "make allfifo" and it will create all fifos needed for 7 switches.
- The program uses "setitimer" and "SIGALRM" to manage "delay" packets
- When a switch or master exits on the other side, pollfds[i].revents will always trigger POLLIN and "read()" reads 0 bytes . Therefore my program detects that in rcvFrame() function and set pollfd[i].fd = -1 so we dont't poll that file descriptor from since that switch/master have terminated.
- When one of the switches terminates, you will see the msg "`Received frame has length= 0 (expected= 28)`" on all of the still-running switches/master. Ignore this, this is just a warning msg from rcvFrame() since the file descriptor of the terminated switch closed.
- I have fds[8][8] that represents an adjacency matrix of file descriptors. Fds[i][j] represents a file descriptor for fifo-i-j
- I have a MASTERSWITCH struct to keep track of master switch info (#hello, #ask, etc)
- I have a SWITCH globalswitch to keep track of pswi info (#hello, #ask, ..etc)
- I have master switch struct and switch struct as global variable, but I still passed them through do_switch() and do_master() because I had them as local variables to main at first. However, I later realized that my signal handler function cannot pass anything in parameters so I had to move them to a global variable last minute.
- **Assumption**: My relay will not relay more than 1 neighbor away.

Project Status: describe the status of your project (to what degree the program works as specified) mention difficulties encountered in the implementation –

My program has matching outputs for ex1.dat, ex2.dat, and ex3.dat.

I think I am finished, except I have not implemented RELAY to relay recursively to many neighbors. my RELAY only relays to the next neighbor.

-I do not have error checks for all the nuances in the command line argument, so please run this program with correct arguments.
-My program successfully ignores line started with '#' when I parse the input file. However, I do not have error checks for invalid headers so make sure the lines in the files abide by the sample input and assignment specification.
-I have not tested for edge cases (I cannot think of any)

Testing and Results: comment on how you tested your implementation, and discuss the obtained results

My program get all the correct output for ex1.dat, ex2.dat, ex3.dat

Acknowledgments: acknowledge sources of assistance:

-sendFrame, rcvFrame, printFrame, compose___() functions are all inspired by the codes from the lab. Aside from sendFrame(), other functions(rcvFrame(), printFrame(), compose__()) are modified by me to meet the A2 specification. THe use of C UNION and ENUM are also inspired by code in the lab.

I copied the FATAL and WARNING function from the lab code but I never used it.

The use of fd[8][8] to represent all the file descriptors are inspired by professor Ehab as he recommended this approach in the lecture.

The use of setitimer() to implement the "delay" package is also inspired by professor Ehab as he recommended this approach in the lecture.