CS 377: Assigment 2, 17 January 2014
Processes, Signals and Shared memory

A process P creates child processes $Q_1 \ldots Q_n$, where the first child process to be created is called Q_1, the next one is Q_2, etc., and fulfills some data processing requirements of the user by using the child processes.

The general idea is that we need to store email addresses, however, each domain (eg gmail.com, outlook.com) is handled by a particular child process Q_i. The process P is reponsible for overall coordination while each child process is responsible for a particular domain and stores all the user IDs corresponding to that domain. These can be maintained in an array like structure and each entry has an index. You may assume the user IDs, domain names and extensions to contain alphanumeric characters, without special characters. The only special characters @ and . demarcate the three components in an email id, e.g., xyz@gmail.com.

P accepts commands from a user and fulfills them by

- Sending instructions to a child process by sending it signals, and

- Sending data to a child process through shared memory.

A child process manages the data items handed to it by P through shared memory and responds to P's queries also by using shared memory.

P must respond to the following commands from the user:

1. *add_email abc@xyz.com:* This command should check if the domain xyz.com already exists. If not, P first creates a child process for this domain. P then sends this email address to the corresponding child (Q_i) via shared memory. Q then checks if the email address already exists. If so, Q puts the following message in shared memory:
   `Child process xyz.com - Email address already exists.`
   else Q puts the message:
   `Child process xyz.com - Email address abc@xyz.com added successfully.` and sends a signal to P to indicate that it has put a message in shared memory and is ready to receive instructions in the form of signals. (Q_i should use the signal SIGUSR1.)

2. *search_email abc@xyz.com:* P should print an error message as below if the domain xyz.com is not created yet.
   `Parent process - Domain does not exist.`
   Otherwise, P sends the email address to the corresponding child process (Q_i) via shared memory. Q_i searches for the email address and sends a signal to P indicating whether the email address exists or not. It also should send the corresponding index of this user ID (abc) in shared memory (essentially index of its position in an array if an array is used)

   P upon receiving the signal, should print one of the following messages appropriately.
   `Parent process - could not find the email address abc@xyz.com`
   or
   `Parent process - found the email address abc@xyz.com at <index>`

3. *delete_email abc@xyz.com :*

   P should print an error message as below if the domain xyz.com is not created yet.
   `Parent process - Domain does not exist.` Otherwise, P sends the email address to the

corresponding child process (Q_i) via shared memory. Q_i searches for the email address and deletes it. Additionally it replies to P with the index of the deleted element using shared memory.

P then prints
`Parent process - child xyz.com could not find the email address abc@xyz.com`
or
`Child process - child xyz.com deleted abc@xyz.com from position <index>.`

4. *delete_domain xyz.com :* P should print an error message as below if the domain xyz.com is not created yet or has already been deleted.
`Parent process - Domain does not exist.`
Otherwise, P then kills the child process (Q_i) corresponding to the domain xyz.com and displays its PID the following.
`Parent process - Domain xyz.com with PID - <PID> deleted.`
Before exiting Q_i prints all the user IDs stored by it.

5. *Quit :* P should terminate all active child processes as in the *kill* command. Before killing a child, all the email addresses of that child must be printed.

Submission: Submit a single tar file per group by 5 pm. The tar file must contain a README file giving names and roll numbers of the group members, and a single file containing your C program.