CSE344 – System Programming - Homework #3
Submission deadline: 23:55, May 13th, 2018

Let's assume for the sake of simplicity that one needs exactly 4 ingredients for preparing a portion of şekerpare:
- eggs
- flour
- butter
- sugar

There are N chefs in a street , and each has an endless supply of two distinct ingredients and lacks the remaining two:

e.g.
chef1 has an endless supply of eggs and flour but lacks butter and sugar,
chef2 has an endless supply of eggs and sugar but lacks flour and butter,
chef3 has an endless supply of butter and sugar but lacks eggs and flour,
etc.

Every chef sits in front of her/his store and waits for the remaining two ingredients to arrive so that she/he can prepare şekerpare.

There is also a wholesaler that every once in a while delivers two random and distinct ingredients out of the 4 to the street of the chefs and then waits for one of the chefs to prepare the şekerpare. We are assuming there is exactly one chef among the N with the right missing ingredients that can prepare the dessert. Once the dessert is ready, the wholesaler takes it for sale and the chefs wait again for the next round.

Your objective is to implement the chefs and the wholesaler as N+1 processes that print on screen their activities. If your program is successful it could for instance print something like the following:

...
chef2 is waiting for flour and butter
chef3 is waiting for eggs and flour
chef1 is waiting for butter and sugar
...
wholesaler delivers eggs and flour
chef3 has taken the eggs
wholesaler is waiting for the dessert
chef3 has taken the flour
chef3 is preparing the dessert
chef3 has delivered the dessert to the wholesaler
wholesaler has obtained the dessert and left to sell it
chef3 is waiting for eggs and flour
...

Rules
a) Your program must handle eventual errors gracefully according to the POSIX traditions.
b) All N+1 processes must terminate in case CTRL-C is sent to any of the N+1 processes.

c) **You are required to use shared memory and semaphores** in order to resolve any and all inter-process communication issues that you may encounter (that means you are not allowed to use busy waiting, or sleep() or pause() or sigsuspend()).

d) Valgrind rule from previous homework still holds and will be hold on every homework.

e) *Your program MUST not give "segmentation fault" or any other fault that causes your program to end in a bad way. It is better for you to send a little part of the homework with clean and executing code instead of full homework with faults like this.*

f) **Provide a makefile to compile your homework. Do not run your program inside makefile. Just compile it.**

) Test your homework using the Virtual Machine given to you.

k) No late submissions.

f) **Do not send any additional files like screenshots. Just send the makefile and source files.**

g) Taking any code from internet will result getting a grade of -100. Do no put links or references to internet, you don't need code from internet to do this homework. Man pages and your book is well enough.

h) Use POSIX and Standard C libraries. You can write your program using C11 standards.

Homework format:
StudentID_HW4_CSE344.tar.gz
|→ Makefile
|→ StudentID_main.c
|→ … (Any other source **files**, not directories!)

Teaching assistant: Ahmet Soyyiğit
e-mail: asoyyigit@gtu.edu.tr

Good luck.