Operating Systems
Tecnológico de Monterrey Campus Chihuahua.

Dr. Alberto Aguilar González

Activity	Activity 04: Augmented LMC.
Mode	2 people team.
Due Date	Sep 10, during session.
Objectives	, Understand processor at an instruction execution level. , Make a "warming" activity , Understand the concept of interrupt.
Description	Your job is to simulate an LMC. The application has the following **minimum requirements**: 1. Able to run programs with the instruction set shown during sessions. 2. Have a GUI that shows, as the program executes, the following: a. Mailboxes content b. Calculator content. c. Calculator flags d. Program counter e. In-box f. Out-box g. Information about the instruction in execution. 3. LMC programs can be executed in two modes: a. Continuous (just pause for INPUT or OUTPUT) b. Step by step (a button to execute one instruction per click) 4. Options to: a. Reset (Program Counter to 0) b. Stop 5. "Interrupt" option. a. After executing an instruction, your application must check if the interrupt flag is set. (This can be implemented as a simple check box that the user can check/uncheck). b. If set, application should ask the user for mailbox number to indicate the address where the interrupt handler resides. Given this, the application executes the interrupt routine which content will be LMC instructions. c. You'll need to add an additional instruction to mark the end of the interrupt handler, we'll name this instruction as RET with a value of 999 d. Note that when an interrupt handler is in execution, content of different LMC components should be updated. e. When the interrupt finishes, LMC should continue with the execution of the next instruction from the main program.Note that it is necessary to save the state of the LMC before entering the interrupt handler and restore information after RET instruction. f. REMARK: Note that it is necessary to disable the

	interrupt flag to avoid the program entering in an infinite loop. 6. Remember that we assume that program and certain data must be stored in the mailboxes prior to running the program. Given this, the user should have an interface to input this information.
Some remaks	, You can use the programming language of your choice. ,
Turn-in details	Turn in the following in a zip file: , Source code , Binary (executable) , User manual , *README.txt* file with technical information such as: o Minimum requirements o Folders content o Any other information needed to setup and run the application