CSE 312 Operating Systems
#HW3

jwan huseein

151044078

interrupt handler:

every flavor of the kernel must include global label called "inturrupt handler" beside the main label .

Whenever the hardware causes an interrupt the function SPIM_timerHandler() in the syscall.cpp file is called .

First and before change the pc to the interrupt handler addr the SPIM_timerHandler() function disables the the interrupts , finding the addr of the interrupt handler by using the function lookup_label ("inturrupt_handle"); then save the current PC in temporary variable .

The interrupt handler in assembly :

first and before change any register's value it loads the value of the register's into a temporary space in data segment then make a syscall with value of 22 which is the system call that load the reg a0 with the id of the running process and the register a1 with the id of the next process that will be running, then set the processes states , and move the value of the registers from the temporary place in data segment to the table , then moves the value of the registers of the next process from the table to the registers

then call "gotonext" syscall which is the system call that copies the value of the PC of the next process to the program counter, and enable the interrupts , so the next process is being executed after this step .