Jesús Wahrman 15-11540

Cesar Rosario 15-11295

Organización del computador

Informe Proyecto 2

El proyecto se divide en dos partes, el instrumentador y el manejador de interrupciones.

Instrumentador:

Para el instrumentador hicimos 3 funciones, una que crea un arreglo en el que guardamos enteros, estos enteros representan que tanto se deben desplazar las instrucciones, es decir si movesArray[i][j] = x, significa que la instrucción numero j del programa numero i debe ser desplazada x words. Para esta función se llevaba un contador de cuantos adds hay antes de cada instrucción y se guardaba este valor en el arreglo. Adicionalmente, esta función al llegar al li $v0 10 la cambia por un break 0x10 y el syscall siguiente por un nop.

La segunda función, usando el arreglo creado en la primera, desplaza las instrucciones y coloca los break 0x20, empezando desde la última instrucción del programa hasta la primera.

Y la tercera función, arregla el offset de los beq para que si estos fueron desplazados o las etiquetas a las que saltan fueron desplazadas, estas salten a donde deben luego de ser desplazadas, para esto también usamos el arreglo de la función 1, si los valores de la dirección de la etiqueta y del beq en ese arreglo son iguales, no pasa nada ya que fueron desplazadas la misma cantidad de instrucciones. Y si el valor es distinto, se añade la resta de la dirección de la etiqueta y la del beq entre 4 al campo del offset del beq.

Manejador de Interrupciones:

Para este, usamos muchos arreglos, cuyos propósitos están explicados en el código.

Para la interrupción del break 0x20, usamos un arreglo llamado addCounter, este lleva la cuenta de cuantos adds han sido ejecutados en cada programa, el manejador de esta interrupción lo que hace es cargar el valor de dicho programa en ese arreglo, sumarle 1 y guardar este nuevo valor.

Para la interrupción del break 0x10, primero usamos un arreglo de unos y ceros, llamado finishedProgram, en este actualizamos la casilla del programa que termino con un 1, para indicar que ya termino su ejecución. Y luego buscamos en este arreglo el siguiente programa no terminado, al conseguirlo cargamos los valores de sus registros que están en registerValues, cargamos la dirección en la que se quedo ese programa que esta en programReturn y ajustamos currentProgram que es un entero que representa el que se esta ejecutando en ese instante, y

saltamos a la dirección en la que se quedo ese programa. Adicionalmente, si ya todos los programas fueron terminados, se imprime cuantos adds ejecuto cada programa con el formato pedido.

Para la interrupción por teclado que pasa al siguiente programa, primero guardamos todos los registros y la dirección en la que se quedo ese programa en registerValues y programReturn, luego buscamos el siguiente programa no terminado, y actualizamos currentProgram, cargamos los valores de sus registros y además la dirección en la que se quedo dicho programa y saltamos a esta.

Para la interrupción por teclado que pasa al programa anterior, es análogo a la anterior pero buscando el programa anterior y no el siguiente.

Y para finalizar la interrupción por teclado que termina la ejecución de los programas, esta va cargando los valores de addCounter y los va imprimiendo con el debido formato.