

Nombre: Guillem Gràcia Andreu

Grupo: 11

Nombre: Gerard Madrid Miró

Hoja de respuesta al Estudio Previo

1. Explicad para qué sirven y qué operandos admiten las instrucciones:

psubb

Instrucción para restar Packed Bytes. Admite dos operandos: dirección origen y destino.

pcmpgtb

Instrucción para comparar Packed Byte (greater than). Admite los dos operandos a comparar. El resultado se almacena en el segundo operando.

movdqa

Instrucción para mover un Double Quadword a una dirección alineada en memoria. Los operandos son origen y destino.

movdqu

Instrucción para mover un Double Quadword a una dirección no alineada en memoria. Los operandos son origen y destino.

emms

Asigne valor vacío al TAG del único operando.
↑
todo unos

2. La propiedad __attribute__ y el atributo aligned sirven para:

--attribute--((aligned)) especifica el alineamiento mínimo para las variables o campos de estructura medido en bytes.

3. Programad en ensamblador una versión de la rutina que hay en `Procesar.c` procurando hacerla lo más rápida posible.

<pre> movl 16(%ebp), %ecx movl 20(%ebp), %edx mull %edx, %edx addl %ecx, %edx for: cmpl %eax, %edx jge endfor movb (%eax), %esi movb (%ebx), %edi subb %esi, %edi cmpb \$0, %edi jge else </pre>	<pre> movb \$255, (%ecx) jmp endif else: movb \$0, (%ecx) endif: incl %eax incl %ebx incl %ecx jmp for </pre>
--	---

4. Explicad como se puede cargar un valor inmediato en un registro xmm usando la instrucción `movdqu`.

Pasamos los words como inmediatos y guardamos en pila. Hacemos movdqu con
@pila y xmm. Finalmente veciamos la pila

5. Programad en ensamblador una versión SIMD de la rutina que hay en `Procesar.c`.

<pre> for: cmpl %eax jge endfor movdqu (%eax), %xmm0 movdqu (%ebx), %xmm1 psubb %xmm0, %xmm1 pcmpgbl \$0, %xmm1 incl %eax incl %ebx incl %ecx jmp for </pre>	
---	--

6. Escribid un código en ensamblador que a partir de un valor almacenado en un registro averigüe si es múltiplo de 16.

<pre> Suponiendo que está en %eax test \$15, %eax jnz noesmultiplo </pre>	
--	--