.file 1 "data\_struct.c"

.section .mdebug.abi32

.previous

.rdata

.align 2

### ↑我覺得是檔案描述

.LC0: # student s1

.word 60

.word 70

.word 70

.align 2

.LC1: # student s2

.word 70

.word 50

.word 80

.text

.align 2

.globl main

.set nomips16

.ent main

main: # main開始

.frame $sp,32,$31 # vars= 32, regs= 0/0, args= 0, gp= 0

.mask 0x00000000,0

.fmask 0x00000000,0

.set noreorder

.set nomacro

addiu $sp,$sp,-32 # 把$sp的資料+(-32)的值寫回$sp中(without overflow)

li $4,536870912 # 0x20000000 -> volatile struct student\* A

ori $6,$4,0x20 # $6 = $4 OR 0x20

lui $2,%hi(.LC0) # upper halfword of $2 = %hi(.LC0) (16-bit number)

lw $9,%lo(.LC0)($2) # $9 = $2 +(shift) lo(.LC0)

addiu $2,$2,%lo(.LC0) # $2 = $2 + %lo(.LC0) (without overflow)

lw $7,4($2) # $7 = $2 shift 4 bit

lw $8,8($2) # $8 = $2 shift 8 bit

lui $2,%hi(.LC1) # upper halfword of $2 = %hi(.LC1) (16-bit number)

lw $5,%lo(.LC1)($2) # $5 = $2 shift %lo(.LC1)

addiu $2,$2,%lo(.LC1) # $2 = $2 + %lo(.LC1)

lw $3,4($2) # $3 = memory($2+4)

lw $2,8($2) # $2 = memory($2+8)

sw $5,16($sp) # memory($sp+16) = $5

sw $3,20($sp) # memory($sp+20) = $3

sw $2,24($sp) # memory($sp+24) = $2

sw $9,0($4) # memory($4) = $9

sw $7,4($4) # memory($s4+4) = $7

sw $8,8($4) # memory($4+8) = $8

lw $3,20($sp) # $3 = memory($sp+20)

lw $4,24($sp) # $4 = memory($sp+24)

lw $2,16($sp) # $2 = memory($sp+16)

sw $2,0($6) # memory($6) = $2

sw $3,4($6) # memory($6+4) = $3

sw $4,8($6) # memory($6+8) = $4

j $31 # go to $31 -> return addr

addiu $sp,$sp,32 # $sp=$sp+32 (without overflow)

.set macro

.set reorder

.end main

.size main, .-main

.ident "GCC: (GNU) 3.4.4 mipssde-6.06.01-20070420"