.file 1 "max\_return.c"

.section .mdebug.abi32

.previous

.text

.align 2

.globl main

.set nomips16

.ent main

# ↑檔案描述

main: # main開始

.frame $sp,24,$31 # vars= 0, regs= 1/0, args= 16, gp= 0

.mask 0x80000000,-8

.fmask 0x00000000,0

.set noreorder

.set nomacro

### ↑檔案描述

addiu $sp,$sp,-24 # $sp = $sp + (-24) (without overflow)

sw $31,16($sp) # memory($sp+16) = $31

li $4,44 # 0x2c $4 = 44 -> int a

li $5,87 # 0x57 $5 = 87 -> int b

jal sum # jump to sum and save next instruction in rd

li $6,2 # 0x2 $6 = 2 -> int c

li $3,536870912 # 0x20000000 -> volatile int\* n

sw $2,0($3) # memory($3) = $2

move $2,$0 #$2 = $0

lw $31,16($sp) $31 = memory($sp+16)

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

addiu $sp,$sp,24 # $sp = $sp + 24 (without overflow)

# main結束

.set macro

.set reorder

.end main

.size main, .-main

.align 2

.globl sum

.set nomips16

.ent sum

sum: # sum func開始

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

.mask 0x00000000,0

.fmask 0x00000000,0

.set noreorder

.set nomacro

addu $2,$4,$5 # $2 = $4+$5 (without overflow)

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

addu $2,$2,$6 # $2 = $2+$6 (without overflow)

# sum func 結束

.set macro

.set reorder

.end sum

.size sum, .-sum

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