Universidade do Minho

Módulo 9

Formato de Instruções: MIPS32

1. Introdução

Pretende-se com esta aula prática que os alunos se familiarizem com o formato de instruções do MIPS32, convertendo um programa de *assembly* para código máquina.

Nota: a questão assinalada com TPC deve ser resolvida **antes da sessão TP** e entregue ao docente no início da mesma. A folha anexa a este módulo, assinalada como TPC, destina-se a esse fim.

---

2. Conversão para binário

Considere o seguinte programa, escrito em C:

prog.c

```
typedef struct {
  char nome[10];
  int idade; } Telem;

Telem membros[10];

main ()
{
  int i, total;

  total = 0;
  for (i=0 ; i< 10 ; i++)
        total += membros[i].idade;
  printf ("%d\n", total);
}
```

Uma vez compilado usando o comando

mips-gcc –S –O1 -mrnames prog.c

gera o código *assembly* apresentado na próxima tabela.

prog.s

```
$LC0: .ascii      “%d\n\000"
main:
      subu  $sp,$sp,24        ####
      sw    $ra,16($sp)       ####
      move  $a1,$zero         ####
      move  $v1,$zero
      la    $a0,membros       ####
      sll   $v0,$v1,4         ####
$L9:
      addu  $v0,$v0,$a0
      lw    $v0,12($v0)       ####
      nop                     ####
      addu  $a1,$a1,$v0
      addu  $v1,$v1,1
      slt   $v0,$v1,10        ####
      .set  noreorder
      .set  nomacro
      bne   $v0,$zero,$L9     ####
      sll   $v0,$v1,4
      .set  macro
      .set  reorder

      lui   $a0,%hi($LC0) # high
      .set  noreorder
      .set  nomacro
      jal   printf
      addiu $a0,$a0,%lo($LC0) # low
      .set  macro
      .set  reorder

      lw    $ra,16($sp)
      nop
      .set  noreorder
      .set  nomacro
      j     $ra
      addu  $sp,$sp,24
      .set  macro
      .set  reorder
      .end main
      .comm membros,160
```

Questão 1 – Passe para código máquina as instruções assinaladas com #### , sabendo que:

- `subu`, `move`, `la` são pseudo-instruções;
- algumas instruções com valores imediatos aparecem com a mnemónica da instrução correspondente do tipo R, mas o opcode apropriado é do tipo I (exemplo: `slt` versus `slti`).
- `nop` consiste numa sequência de 32 bits a 0;
- os endereços de `membros`, `$LC0` e `printf` não são conhecidos em tempo de compilação; devem ser considerados 0x00000000, com 32 bits de tamanho;
- os números dos registos são dados na próxima tabela:

Nome	Número	Utilização	Preservado na chamada?
$zero	0	Constante 0	n.a.
$v0-$v1	2-3	Valores para resultados e avaliação de expressões	Não
$a0-$a3	4-7	Argumentos	Sim
$t0-$t7	8-15	Temporários	Não
$s0-$s7	16-23	Seguros	Sim
$t8-$t9	24-25	Mais temporários	Não
$gp	28	Apontador global	Sim
$sp	29	Apontador para pilha	Sim
$fp	30	Apontador para a *frame*	Sim
$ra	31	Endereço de retorno	Sim

Verifique as suas respostas usando os comandos

```
mips-gcc –O1 –c prog.c
mips-objdump –d prog.o
```

TPC	
Número:	Turno:
Nome:	

TPC : Apresente, em binário e hexadecimal, o código da instrução `sw $ra,16($sp)`, justificando todos os dados.