Tugas Projek (KAT-3: 15%) OS, semester genap T.A. 2016/2017

Semua persoalan di bawah ini murni hanya program yang berkaitan dengan memori dan prosesor saja, dan belum (tidak) melibatkan elemen computer: input, output, maupun file.

Coba: tempatkan program ke lokasi bukan 0, atau perintahkan eksekusi pada lokasi yang ditempatkan.

0. Jalankan contoh program (sample03, sample04, dan sample05), dan pelajari perilaku program, penempatan program di memori, dan perubahan isi memori, perubahan isi register, dan aktifitas eksekusi eksekusi instruksi demi instruksi.

Dari soal 1 hingga seterusnya dikerjakan dalam kelompok, maksimum 2 orang per kelompok.

1. Mengalihkan contoh source program yang diberikan (pseudocode atau dalam bahasa C/C++) ke dalam kode bahasa mesin yang menggunakan nama-nama pada instruction set. Dan, dilanjutkan dialihkan ke dalam nilai-nilai instruksi yang dapat dieksekusi oleh computer.

void main()

{ int a[5] = {3, 12, 4, 7, 1}; // int: numerik 32 bits

int b[5];

int k;

for (k=0; k<5;k++) {

b[k] = a[k];

} // endfor-k

}

Variabel adalah lokasi memori. Dalam implementasi ke dalam bahasa mesin, variabel *k* anda dapat pilih diimplementasi sebagai register, atau tetap menggunakan memori. Pilihan tersebut perlu memperhatikan implikasinya, yaitu jika variabel *k* menggunakan memori, maka setiap kali eksekusi instruksi, anda harus mengupdate (memperbaharui) lokasi memori tersebut.

2. Membuat sendiri minimal dua source program berbeda yang sudah teralihkan ke dalam nilai-nilai instruksi yang dapat dieksekusi oleh komputer. Banyaknya instruksi dari tiap program minimal 20 baris instruksi bahasa mesin. Petunjuk bantuan: mulailah dari membuat ide program dalam bentuk pseudocode atau dalam bentuk bahasa C/C++ terlebih dahulu. Dari ide program tersebut alihkan satu tahap demi satu tahap, hingga mendapatkan bentuk program bahasa mesin (nilai instruksi) yang dapat dieksekusi computer.

3. *Membuat software untuk mengalihkan program* yang ditulis dalam kode bahasa mesin (menggunakan nama instruction set) ke dalam nilai-nilai instruksi yang dimengerti oleh prosesor dan control unit. Pada source program yang berisi kode dalam bahasa mesin sudah menyertakan alamat memori dari program yang bersangkutan. Contoh program dapat dilihat pada lampiran.

4. Membuat program yang mengelola (semacam dan perluasan versi cekcpu.jav) atau memanage permintaan eksekusi program dari user. Peran tugas yang dilakukan adalah menerima sejumlah file program yang executable untuk dieksekusi oleh computer. Setelah sedikitnya dua program tersebut diterima dan ditempatkan di memori, maka eksekusi dilakukan, asumsi program-program tersebut bersifat independen yaitu tidak saling terkait (berkomunikasi). Pola eksekusi yang dicoba (1) satu program selesai dieksekusi dilanjutkan dengan program berikutnya hingga semua program dieksekusi, (2) program dieksekusi bergantian dengan batas waktu tertentu, sehingga semua program tersebut selesai dieksekusi. Petunjuk bantuan, perhatikan lokasi penempatan setiap program terhadap alamat memori (yang dinyatakan melalui immediate address maupun alamat address yang disimpan oleh register).

5. [p] Lanjutan

Pseudocode: sample03

Catatan:

rx – r adalah register, dan x adalah identitas register

m[x] – m adalah lokasi memori, x adalah alamat fisik memori

r1 = 25

r3 = 75

r2 = r1+r3

r4 = 50

r2=r4+r2

r5 = 19

r2 = r5 + r2

r2 = r2 + 5

r6 = r2

m[r7+128] = r2

r0 = m[r7+128]

m[r7+132] = r0

r9 = m[r7+132]

goto p

exit: end

p: if (r2 != r9) goto vtrue

r12 = 5

goto endif

vtrue: r12 = 8

endif: r11 = 56 ;alamat exit

goto exit

sample03: low level program

|  |  |  |
| --- | --- | --- |
| Pseudocode  r – register, m - memory | Alamat memori program, instruction set | Instruksi dalam bahasa mesin (instruction set) |
| r1 = 25  r3 = 75  r2 = r1+r3  r4 = 50  r2=r4+r2  r5 = 19  r2 = r5 + r2  r2 = r2 + 5  r6 = r2  m[r7+128] = r2  r0 = m[r7+128]  m[r7+132]:0-7 = r0 ; byte  r9 = m[r7+132]:0-7 ; byte  goto p  exit: end  p: if (r2 != r9) goto vtrue  r12 = 5  goto endif  vtrue: r12 = 8  endif: r11 = 56 ;alamat exit  goto exit | [00] movi r1,25  [04] movi r3,75  [08] add r2,r1,r3  [12] movi r4,50  [16] add r2,r2,r4  [20] movi r5,19  [24] add r2,r5,r2  [28] addi r2,r2,5  [32] movr r6,r2  [36] mov [r7+128],r2  [40] movm r0,[r7+128]  [44] movb [r7+132],r0  [48] movmb r9,[r7+132]  [52] jmp 60  [56] halt  [60] jne r2,r9,72  [64] movi r12,5  [68] jmp 76  [72] movi r12,8  [76] movi r11,56  [80] jmp r11 [00] | [00] 0x03100019 = 51380249  [04] 0x0330004B = 53477451  [08] 0x00213000 = 2174976  [12] 0x03400032 = 54526002  [16] 0x00242000 = 2367488  [20] 0x03500013 = 55574547  [24] 0x00252000 = 2433024  [28] 0x01220005 = 19005445  [32] 0x02620000 = 39976960  [36] 0x04720080 = 74580096  [40] 0x05070080 = 84344960  [44] 0x06700084 = 108003460  [48] 0x07970084 = 127336580  [52] 0x1500003C = 352321596  [56] 0x7F000000 = 2130706432  [60] 0x17290048 = 388563016  [64] 0x03C00005 = 62914565  [68] 0x1500004C = 352321612  [72] 0x03C00008 = 62914568  [76] 0x03B00038 = 61866040  [80] 0x16B00000 = 380633088 |

Latihan:

Ubahlah pseudocode berikut ini menjadi machine code (r – register, m – lokasi memori)

Latihan-01: sample04

r1 = 0

r2 = 10;

While (r1 < 10)

r2 = r2 + r1

r1 = r1 +1

Endwhile

End program

Solusi: sample04

|  |  |  |
| --- | --- | --- |
| Pseudocode | Assembly code | Address code |
| r1 = 0  r2 = 10;  While (r1 < 10)  r2 = r2 + r1  r1 = r1 +1  Endwhile  End program | movi r1, 0  movi r2, 10  movi r3, 10  ulang: je r1, r3, endprogram  jgt r1, r3, endprogram  add r2, r2, r1  addi r1, r1, 1  jmp ulang  endprogram: halt | [00] movi r1, 0  [04] movi r2, 10  [08] movi r3, 10  [12] je r1, r3, 32  [16] jgt r1, r3, 32  [20] add r2, r2, r1  [24] addi r1, r1, 1  [28] jmp 12  [32] halt |
| Machine code: (instr. set), H | Machine code: decimal |  |
| [00] 0x03100000  [04] 0x0320000A  [08] 0x0330000A  [12] 0x17130020  [16] 0x1A130020  [20] 0x00221000  [24] 0x01110001  [28] 0x1500000C  [32] 0x7F000000 | [00] 51380224  [04] 52428810  [08] 53477386  [12] 387121184  [16] 437452832  [20] 2232320  [24] 17891329  [28] 352321548  [32] 2130706432 |  |

**Catatan**: tambahan r3 (atau r-lainnya) digunakan untuk mengakomodasi sintaks je dan jgt, karena tidak memperkenankan konstanta

Latihan-02: sample05

r1 = 0

r2 = 10;

While (r1 < 10)

r2 = r2 + r1

r1 = r1 +1

Endwhile

*a = r2 ; a – merupakan variabel, maka menempati ruang memori, diasumsikan lokasi 64*

End program

Solusi: sample05

Variable a, adalah lokasi memori dengan alamat 64 (sebagai contoh)

|  |  |  |
| --- | --- | --- |
| Pseudocode | Assembly code | Address code |
| r1 = 0  r2 = 10;  While (r1 < 10)  r2 = r2 + r1  r1 = r1 +1  Endwhile  a = r2  End program | movi r1, 0  movi r2, 10  movi r3, 10  ulang: je r1, r3, simpan  jgt r1, r3, simpan  add r2, r2, r1  add r1, r1, 1  jmp ulang  simpan: movi r3,64  mov [r3+0],r2  endprogram: halt | [00] movi r1, 0  [04] movi r2, 10  [08] movi r3, 10  [12] je r1, r3, 32  [16] jgt r1, r3, 32  [20] add r2, r2, r1  [24] addi r1, r1, 1  [28] jmp 12  [32] movi r3,64  [36] mov [r3+0],r2  [40] halt |
| Machine code: (instr. set), H | Machine code: decimal |  |
| [00] 0x03100000  [04] 0x0320000A  [08] 0x0330000A  [12] 0x17130020  [16] 0x1A130020  [20] 0x00221000  [24] 0x01110001  [28] 0x1500000C  [32] 0x03300040  [36] 0x04320000  [40] 0x7F000000 | [00] 51380224  [04] 52428810  [08] 53477386  [12] 387121184  [16] 437452832  [20] 2232320  [24] 17891329  [28] 352321548  [32] 53477440  [36] 70385664  [40] 2130706432 |  |

**Catatan**: pada movi r3,64 angka 64 (diasumsikan, anda dapat menentukan sendiri, asalkan tidak menggunakan lokasi memori yang digunakan oleh program) menunjukkan alamat memori pada yang merepresentasikan atau merupakan lokasi dari variabel a.

Latihan-02: sample05

r1 = 0

r2 = 10;

While (r1 < 10)

r2 = r2 + r1

r1 = r1 +1

Endwhile

*a = r2 ; a – merupakan variabel, maka menempati ruang memori, diasumsikan lokasi 64*

End program

Solusi: sample05-a

Variable a, adalah lokasi memori dengan alamat 64 (sebagai contoh)

|  |  |  |
| --- | --- | --- |
| Pseudocode | Assembly code | Address code |
| r1 = 0  r2 = 10;  While (r1 < 10)  r2 = r2 + r1  r1 = r1 +1  Endwhile  a = r2  r2 = 0xA527B  b = r2:[16-31]  d = r2:[0-15]  r5 = d ; data 16 bits  End program | movi r1, 0  movi r2, 10  movi r3, 10  ulang: je r1, r3, simpan  jgt r1, r3, simpan  add r2, r2, r1  add r1, r1, 1  jmp ulang  simpan: movi r3,128  mov [r3+0],r2  movi r2,0xA527B  movh [r3+4],r2  movl [r3+6]],r2  movmhw r5,[r3+6]  endprogram: halt | [00] movi r1, 0  [04] movi r2, 10  [08] movi r3, 10  [12] je r1, r3, 32  [16] jgt r1, r3, 32  [20] add r2, r2, r1  [24] addi r1, r1, 1  [28] jmp 12  [32] movi r3,64  [36] mov [r3+0],r2  [40] movi r2,0xA527B  [44] movh [r3+4],r2  [48] movl [r3+6]],r2  [52] movmhw r5,[r3+6]  [56] halt |
| Machine code: (instr. set), H | Machine code: decimal |  |
| [00] 0x03100000  [04] 0x0320000A  [08] 0x0330000A  [12] 0x17130020  [16] 0x1A130020  [20] 0x00221000  [24] 0x01110001  [28] 0x1500000C  [32] 0x03300080  [36] 0x04320000  [40] 0x032A527B  [44] 0x08320004  [48] 0x0A320006  [52] 0x1D530006  [56] 0x7F000000 | [00] 51380224  [04] 52428810  [08] 53477386  [12] 387121184  [16] 437452832  [20] 2232320  [24] 17891329  [28] 352321548  [32] 53477504  [36] 70385664  [40] 53105275  [44] 137494532  [48] 171048966  [52] 491978758  [56] 2130706432 |  |

**Catatan**: pada movi r3,64 angka 64 (diasumsikan, anda dapat menentukan sendiri, asalkan tidak menggunakan lokasi memori yang digunakan oleh program) menunjukkan alamat memori pada yang merepresentasikan atau merupakan lokasi dari variabel a.

Latihan-03: ide program dalam bahasa C/C++ (tanpa menyertakan interaksi input, dan output device)

Tujuan: mengalihkan high level ke bahasa mesin

G