



CO2008 - KIẾN TRÚC MÁY TÍNH

Khoa Khoa học và kỹ thuật máy tính
Đại học Bách Khoa - ĐHQG Tp.HCM

09/2024

Bài thực hành 4

CHƯƠNG 2 KIẾN TRÚC TẬP LỆNH MIPS: Gọi hàm (lập trình cấu trúc), thời gian thực thi.

Mục tiêu

- Chuyển từ ngôn ngữ cấp cao (C) sang hợp ngữ MIPS.
- Sử dụng lệnh điều khiển (nhảy, rẽ nhánh) để lập trình cấu trúc.
- Biết nguyên lý gọi hàm. Sử dụng các lệnh gọi hàm jal, jr.
- Cách dịch lệnh nhảy j label.

Yêu cầu

- Xem lại cách dùng các lệnh jal, jr.

Bài tập và thực hành

Sinh viên chuyển chương trình C bên dưới qua hợp ngữ MIPS tương ứng.

Bài 1. Leaf function (hàm lá).

Chuyển thủ tục "reverse" (đảo thứ tự chuỗi) từ ngôn ngữ C sang hợp ngữ MIPS. Thủ tục **reverse** được gọi khi thực thi lệnh **jal reverse** từ vùng **.text**. Tham số vào (input) gồm \$a0 chứa địa chỉ biến chuỗi **cArray** (chính là địa chỉ đầu chuỗi cần đảo thứ tự), \$a1 chứa chiều dài chuỗi. Trị trả về (output, nếu có) chứa trong \$v0.

```
char[] cArray="Computer Architecture 2024."
int cArray_size=27;
void reverse(char[] cArray, int cArray_size)
{ int i; char temp;
  for (i=0;i<cArray_size/2;i++)
  { temp=cArray[i];
    cArray[i]=cArray[cArray_size-1-i];
    cArray[cArray_size-1-i]=temp;
  }
}
```

Bài 2. Non-leaf function (là hàm/thủ tục có thực hiện gọi một hàm/thủ tục bên trong).

Chuyển hàm **range** sau đây sang hợp ngữ MIPS.

```
int iArray[10];
int iArray_size=10;
int range(int iArray[], int iArray_size)
{ int temp1,temp2;
  temp1=max(iArray,iArray_size);
  temp2=min(iArray,iArray_size);
  return temp1-temp2;
}
```

Chương trình bắt đầu từ vùng **.text**, sau đó gọi hàm **range**. Trong hàm **range** lại gọi 2 hàm con là **max_day** và **min_day**. Giả sử địa chỉ đầu dãy và số phần tử được gán lần lượt vào các thanh ghi \$a0, \$a1. Xuất giá trị range ra màn hình.

Lưu ý: Khi viết hàm không lá, sinh viên cần thực hiện lưu thanh ghi \$ra vào stack khi bắt đầu hàm như sau:

```
addi $sp,$sp,-4      # adjust stack pointer for one 4-bytes item
sw $ra,0($sp)        # push return address to stack
và phục hồi thanh ghi $ra từ stack trước khi trở về.
lw $ra,0($sp)        # pop return address from stack
addi $sp,$sp,4       # restore stack pointer
```

Bài 3. Cho đoạn code hợp ngữ MIPS bên dưới:

```
        addi $a0,$zero,100    # upper threshold
        addi $a1,$zero,0      # count variable
        add  $a2,$zero,$zero  # sum initialization

loop:
        beq  $a0,$a1,exit
        add  $a2,$a2,$a1
        addi $a1,$a1,1
        j    loop

exit:
```

- Xác định giá trị của thanh ghi \$a2 sau khi thực thi đoạn code trên.
- Xác định tổng số chu kỳ thực thi khi chạy đoạn chương trình trên. Giả sử CPI của các lệnh là 1.
- Đánh đoạn code trên vào MARS MIPS và xác định mã máy của lệnh `j loop`.