Universidade Federal de Ouro Preto

Departamento de Computação e Sistemas (DECSI)

Organização e Arquitetura de Computadores I

Trabalho Prático I – Prof. Vicente Amorim

O que é?

O trabalho mais simples que pode ser executado por um computador são as operações lógicas e aritméticas. Através delas é possível executar as operações mais simples requeridas por qualquer linguagem de programação. Dessa forma, nada mais normal que essas sejam também a base das linguagens de programação de baixo-nível.

De maneira geral, se analisarmos então essas operações logo será possível notar que um computador nada mais é que uma grande calculadora com capacidade de tomada de decisões. As operações mais básicas (soma, subtração, multiplicação e divisão) são comumente já suportadas pela maioria das linguagens.

O que deve ser feito?

Considerando os pontos anteriormente listados, seu objetivo nesse trabalho prático é a construção de uma calculadora que possua uma interface gráfica para interação com o usuário. No mínimo as operações básicas devem ser suportadas (soma, subtração, multiplicação e divisão).

A forma como os dados são solicitados ao usuário, processados e apresentados de volta é inteiramente inerente a cada uma das implementações.

O que deve ser entregue?

1) Documentação descrevendo a lógica principal da sua implementação da calculadora;
2) Código do fonte do seu programa em *assembly;*
3) Documentação e código-fonte devem ser enviados como anexo para: vicente.amorim.ufop@gmail.com. Título do e-mail deve estar no formato:

[OACI][<Numero_matricula_1,Nome_aluno_1,Numero_matricula_2,Nome_aluno_2>][TPI]

Quem? Quando? Valor?

1) Quem: Grupos de 2 alunos.
2) Quando: Entregas até o dia 16/05/2014 (23:59).
3) Valor: 15Pts.
4) Bônus: **+3Pts**.
    a. Soluções mais criativas.

Restrições:

Atrasos:

a. **Notas = 15/(n+1)**,

Onde n: Número de semanas de atraso;

1 semana atraso == (1minuto <= **X** <= 7 dias).

Cópias: Trabalhos não serão considerados e nota zero atribuída.

Dicas

1) Utilize o simulador MARS para executar e testar seu programa. Nele é possível acompanhar passo-a-passo a execução do algoritmo, além de saber o valor de cada um dos registradores. (http://courses.missouristate.edu/kenvollmar/mars/).
2) Listagem das chamadas de sistema disponíveis no MARS: http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html.
3) De modo a facilitar seu trabalho, codifique primeiro o programa em linguagem de alto-nível e depois simplesmente o traduza para *assembly* MIPS.
4) Utilize como referência os trechos de código abaixo. Eles podem ser úteis na leitura dos dados digitados pelo usuário no MARS.

```
.data
str_num:      .asciiz    "Quantos numeros deseja inserir? (maximo 200)\n"

.text
.globl main
main:
  la $a0, str_num
  li $v0, 4
  syscall
  li $v0,5
  syscall
```

Referências

[1] MIPS Instruction set. http://en.wikipedia.org/wiki/MIPS_instruction_set

[2] PATTERSON, D. A., HENNESSY, J. L., Organização e Projeto de Computadores: A Interface Hardware/Software, Ed. Campus, 3ª ed., 2005.