

DEPARTAMENTO DE COMPUTAÇÃO (DECOM)
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I
Professor: Mateus Felipe Tymburibá Ferreira

AULA 4: IMPLEMENTAÇÃO DE FUNÇÕES EM ASSEMBLY DO MIPS

Orientações gerais:

1. Esta atividade deve ser realizada em dupla.
2. **Prazo de entrega do relatório: até 13:00h do dia da próxima aula (06/05/2022).** Enviar pelo link disponível na página da disciplina (no AVA). **Não será permitida a submissão de relatórios após o prazo acima.**
3. Na aula seguinte, o relatório deverá ser brevemente (em aproximadamente 1 minuto) explicado ao professor por pelo menos um aluno da dupla. **Duplas que não comparecerem na aula seguinte para explicar seus relatórios concorrerão a apenas 75% dos pontos previstos para a atividade.**

O que deve ser entregue

- Arquivo PDF nomeado: **"Aula-4-Lab-AOC1_nome-completo-aluno1_nome-completo-aluno2.pdf"**.
- Este arquivo deverá conter as discussões referentes aos itens a serem respondidos, elencados abaixo. Inclua também em seu relatório, na forma de apêndice, o código Assembly implementado por você.
- Escreva em texto simples e conciso as suas análises e considerações.
- Responda às perguntas realizadas na prática, quando for o caso.

O que deve ser feito

- 1) Abra o simulador Mars (acesse¹ para tutorial e download): `java -jar Mars4_5.jar`
- 2) Crie um código que implemente em Assembly do MIPS o seguinte código em C:

```
int f(int y){
    int i;
    int a[32] = {-1,-1};
    for(i = 2; i <= y; i++){
        a[i] = a[i-1] + a[i-2];
    }
    return(a[y]);
}

int main(){
    int result = f(32);
    printf("Valor retornado: %d\n", result);
    return result;
}
```

O que deve ser respondido

- 1) Explique o quê o código da função "f" faz.
- 2) O que acontece se o valor de y for 32? Qual valor é retornado nesse caso? Aponte os valores intermediários assumidos por "a[i]" e por "i" durante uma chamada com esse valor (f(32)).
- 3) O que poderia acontecer se o prólogo da função "f" guardasse o seu endereço de retorno na pilha (para poder efetuar uma chamada a outro procedimento, por exemplo), no primeiro espaço disponível no "frame" da função?

¹ <http://courses.missouristate.edu/KenVollmar/MARS/>

Sugestões:

1) Para facilitar a observação do evento ocorrido na chamada `f(32)`, atualize o valor em memória das variáveis (via instrução “store” / `sw`) sempre que o conteúdo do registrador correspondente for alterado, e carregue (via instrução “load” / `lw`) o valor atualizado em memória para o registrador correspondente à variável, sempre que o conteúdo da variável for usado.

2) Altere o valor do parâmetro de chamada da função “f” para um valor maior que 34 (exemplo: `f(35)`). Em uma máquina Linux atual (exemplo: Ubuntu 20.04) com uma versão recente do GCC (versão próxima ou superior à 9.4.0), compile o código com a opção `-O0` e execute-o. Observe o resultado! Depois disso, adicione a seguinte opção de compilação: `-fno-stack-protector` . Execute novamente o programa e observe o resultado. Pesquise na Internet o significado das mensagens exibidas ao executar o programa, bem como a função da opção de compilação “`fno-stack-protector`” e anote suas conclusões.