

AOC-1: Trabalho Prático 1

► Instruções

- Use apenas instruções vistas em aula até agora (slides)
- No MARS, use a seguinte configuração:
 - No menu *Settings*, desabilite as opções:
 - *Permit extended (pseudo) instructions and formats* e
 - *Delayed Branching*
- Seus exercícios serão corrigidos com o MARS configurado da forma descrita acima
- Utilize **EXATAMENTE** os registradores explicitados nos exercícios
 - Resultados armazenados em registradores diferentes serão considerados incorretos
- Comente seu código
- Todos os exercícios são individuais
 - Cópias detectadas resultarão em nota zero para ambos os alunos

AOC-1: Trabalho Prático 1

► Instruções

- O material deve ser entregue pelo AVA (<http://ava.ufpel.edu.br>) e deverá obedecer às seguintes regras:
 - Será um arquivo compactado (**obrigatoriamente** no formato **.zip**) contendo os códigos fonte dos TPs
 - Ex: **nome_aluno.zip**
 - Cada exercício deve ter o seguinte nome:
 - **matricula_tp{n}_e{m}.asm**
 - Onde:
 - **matricula** é a matrícula do aluno e
 - **{n}** é o número do TP
 - **{m}** é o número do exercício
 - Ex: **16100001_tp1_e1.asm**, **16100001_tp1_e2.asm**, ...
- Trabalhos que não seguirem as regras acima **NÃO SERÃO CORRIGIDOS!**
- O prazo de submissão do trabalho é até as **23:55** da próxima **terça-feira**. **NÃO** serão aceitos exercícios após a data/hora-limite.

AOC-1: Trabalho Prático 1

► Exercícios

1. Faça um programa que escreva o valor **0xDECADE70** no registrador **\$t7**, incluindo um dígito hexadecimal por vez (isto é, insira letra por letra, individualmente) no registrador.
2. Escreva **0xAAAAAAAA** em **\$t1**. Faça um *shift* de um bit para a direita desse valor e coloque o resultado em **\$t2**, deixando **\$t1** inalterado. Em **\$t3**, **\$t4** e **\$t5** coloque os resultados das operações **or**, **and** e **xor** entre **\$t1** e **\$t2**, respectivamente. Em comentários no final do código, explique os resultados obtidos, mostrando os valores binários.

AOC-1: Trabalho Prático 1

► Exercícios

3. Comece um programa pela instrução: **ori \$t1, \$zero, 0x01**.
Usando apenas as instruções lógicas do tipo R (dois registradores como operandos) or, and e xor e instruções de *shift* (isto é, sem usar outras instruções ori além da inicial e nem instruções nor), escreva **0xFFFFFFFF** em **\$t1**. Procure usar o menor número de instruções possível.
4. Escreva a sequência 0x12345678 em \$t1. A seguir, escreva um código que inverta a sequência, escrevendo 0x87654321 em \$t2. Obviamente o código deve inverter os bits de \$t1 e não simplesmente escrever 0x87654321 diretamente em \$t2.