

UNIVERSIDADE ESTADUAL DE MARINGÁ
DEPARTAMENTO DE INFORMÁTICA

PROGRAMAÇÃO PARA INTERFACEAMENTO DE HARDWARE E SOFTWARE – PIHS

Relatório Tarefa 05 – Partes 3 e 4

(baseado no material do Prof. Ronaldo Augusto)

Tarefa:

- Nesta tarefa deve ser entregue **capturas da tela de execução do código com comentários** (arquivo .pdf)
 - Utilize o **GNU Debugger** para verificar as alterações realizadas nos registradores e variáveis.
1. Execute o teste 17 que utiliza registradores de 32, 16 e 8 bits e demonstre o funcionamento da **instrução inc**, conferindo os valores dos registradores durante e após a execução do código.
 2. Execute o teste 18 que utiliza registradores de 32, 16 e 8 bits e demonstre o funcionamento da **instrução dec**, conferindo os valores dos registradores durante e após a execução do código.
 3. Execute o teste 19 que utiliza registradores de 64 bits (concatenando EDX e EAX) e demonstre o funcionamento da **instrução divl**. Após a execução da instrução, **quais os registradores que armazenam o resultado** (quociente e resto)?
 4. Execute o teste 20 que utiliza registradores de 64 bits (dividendo natural e divisor negativo) e demonstre o funcionamento da **instrução idivl**. Após a execução da instrução, **quais os registradores que armazenam o resultado** (quociente e resto)? Porque não foi preciso fazer ajuste nos registradores EDX e EAX?
 5. Execute o teste 21 que utiliza registradores de 64 bits (dividendo negativo e divisor natural) e demonstre o funcionamento da **instrução idivl**. Após a execução da instrução, **quais os registradores que armazenam o resultado** (quociente e resto)? **Explique** o ajuste nos registradores EDX e EAX com relação a representação de números negativos e a necessidade de subtrair o valor 1 (demonstrando o valor do registrador antes e depois da manipulação).
 6. Execute o teste 22 que utiliza registradores de 32 bits e demonstre o funcionamento da **instrução divl**. Mostre os registradores com o resultado.

7. Execute o teste 23 que utiliza registradores de 32 bits (dividendo natural e divisor negativo) e demonstre o funcionamento da **instrução idivl**. Mostre os registradores com o resultado.
8. Execute o teste 24 que utiliza registradores de 32 bits (dividendo negativo e divisor natural) e demonstre o funcionamento da **instrução idivl**. Mostre os registradores com o resultado. **Explique** de forma demonstrativa a instrução **cdq**. Faça uma alteração no código inicializando o EDI com -1 (FFFF FFFF) e teste.
9. Execute o teste 25 que divide um dado 32 bits por um dado de 16 bits e demonstre o funcionamento da **instrução divw**. Após a execução da instrução, **quais os registradores que armazenam o resultado** (quociente e resto)?
10. Execute o teste 26 que divide um dado 16 bits por um dado de 8 bits e demonstre o funcionamento da **instrução divb**. Após a execução da instrução, **quais os registradores que armazenam o resultado** (quociente e resto)? **Explique as instruções sarw e andw** no contexto deste teste.
11. Execute o teste 27 que multiplica um dado de 32 bits por um dado de 32 bits (inteiros sem sinal) e demonstre o funcionamento da **instrução mull**. Após a execução da instrução, **quais os registradores que armazenam o resultado**? Explique.
12. Execute o teste 28 que multiplica um dado de 16 bits por um dado de 16 bits (inteiros sem sinal) e demonstre o funcionamento da **instrução mulw**. Após a execução da instrução, **quais os registradores que armazenam o resultado**? Explique.
13. Execute o teste 29 que multiplica um dado de 8 bits por um dado de 8 bits (inteiros sem sinal) e demonstre o funcionamento da **instrução mulb**. Após a execução da instrução, **quais os registradores que armazenam o resultado**? Explique.
14. Utilize as instruções **negb**, **negw** ou **negl** em um dos testes já realizados para demonstrar o seu uso.