



LINGUAGENS DE MONTAGEM

Estudo dirigido 1

Unioeste - 2º. Ano de CC

Prof. Newton Spolaôr

Avisos Preliminares

- Para cada tarefa indicada neste estudo dirigido são solicitadas atividades variadas
- Caso você tenha dificuldade para realizar ou entender alguma tarefa, procure o professor pessoalmente, por *chat* Teams ou email (newton.spolaor@unioeste.br), com a maior antecedência que você puder
- O estudo dirigido vale frequência para seis aulas de 50 minutos, sendo duas em cada data que segue: 22/11, 29/11 e 6/12/24

Avisos Preliminares

- O estudo dirigido deve ser realizado em equipe de 1 ou 2 alunos (preferencialmente 2 alunos)
- O prazo de entrega de todas as tarefas do estudo dirigido é 21/12/24, 23:59 hs
- A forma de entrega do estudo dirigido é por *chat* individual com o professor, a ser enviado por um dos integrantes da equipe
- O plágio entre equipes não é permitido
- Os nomes dos alunos da equipe devem ser informados durante a entrega

Tarefa 1

1. Instale o ambiente Code::Blocks versão 20.03 no seu computador – preferencialmente, com sistema operacional Windows
2. Execute as atividades indicadas no PDF da Aula 2 disponível na equipe do Teams (pasta Material de Aula/Slides), páginas 8 a 28, e estude o conteúdo dessas páginas; naturalmente, os endereços de memória no seu computador podem ser diferentes dos indicados na aula, o que não é um erro

Tarefa 1

3. Registre *prints/screenshots* do Code::Blocks instalado pela equipe em que, além do dia e horário do relógio do sistema operacional do computador, apareça pelo menos
 - O código C do exemplo da aula
 - O código Assembly correspondente ao código C
 - O código resultante do depurador (*disassembly*)
 - O resultado de *memory dump*
4. Acrescente “Tarefa 1” em parte do nome de cada *print/screenshot* registrado nesta tarefa

Tarefa 2

1. Estude os exemplos indicados no PDF da Aula 2 disponível na equipe do Teams (pasta Material de Aula/Slides), páginas 33 a 46
2. Abra a ferramenta *online* Compiler Explorer (<https://godbolt.org>), escolhendo...
 - Compilador para linguagem C na janela da esquerda
 - Depurador para plataforma x86-64 gcc 11.2 com arquitetura de 32 bits (opção `-m32`) na janela da direita
3. Crie um código C original (*int main*), inexistente na Aula 2, com operações aritméticas e de atribuição usando ao menos três variáveis e duas constantes; não defina funções nem use funções prontas do C

Tarefa 2

4. Registre *prints/screenshots* do Compiler Explorer em que apareça pelo menos
- O código C do exemplo da aula
 - O código Assembly correspondente ao código C e que está associado à lógica principal
 - O código resultante do depurador (*compile to binary object*) e que está associado à lógica principal
 - O código binário (em hexadecimal) correspondente à lógica principal, como ilustrado a seguir

y += x;	mov	edx, DWORD PTR y	mov	edx, DWORD PTR ds:0x804c020	8b 15 20 c0 04 08
	mov	eax, DWORD PTR x	mov	eax, ds:0x804c01c	a1 1c c0 04 08
	add	eax, edx	add	eax, edx	01 d0
	mov	DWORD PTR y, eax	mov	ds:0x804c020, eax	a3 20 c0 04 08

Tarefa 2

5. Identifique e informe as seguintes informações
 - Todas as instruções Assembly empregadas
 - Todos os endereços de memória associados a variáveis
 - Todos os registradores usados por instruções Assembly
 - Todas as constantes usadas
6. Pesquise em referências como o material didático dos profs. Murilo Bresciani de Carvalho e Carmen Maria Costa de Carvalho o significado de cada instrução Assembly distinta identificada no passo anterior
7. Descreva a pesquisa do passo anterior em um documento textual

Tarefa 2

8. Acrescente “Tarefa 2” em parte do nome de cada *print/screenshot* e documento gerado nesta tarefa

Tarefa 3

1. Crie um código C original (*int main*), inexistente na Aula 2 e distinto do código C construído na Tarefa 2, com operações aritméticas e de atribuição usando ao menos cinco variáveis e cinco constantes; não defina funções nem use funções prontas do C
2. Realize os Passos 4, 5, 6 e 7 da Tarefa 2 sobre o código que foi gerado no Passo 1 da Tarefa 3; quando uma instrução Assembly identificada na Tarefa 3 já tiver sido usada na Tarefa 2, não é necessário descrevê-la novamente
3. Acrescente “Tarefa 3” em parte do nome de cada *print/screenshot* e documento gerado nesta tarefa

Tarefa 4

1. Crie um código C original (*int main*), inexistente na Aula 2 e distinto do código C construído na Tarefa 2, com operações lógicas e de atribuição usando ao menos três variáveis; não defina funções nem use funções prontas do C
2. Realize os Passos 4, 5, 6 e 7 da Tarefa 2 sobre o código que foi gerado no Passo 1 da Tarefa 4; quando uma instrução Assembly identificada na Tarefa 4 já tiver sido usada nas Tarefas 2 ou 3, não é necessário descrevê-la novamente
3. Acrescente “Tarefa 4” em parte do nome de cada *print/screenshot* e documento gerado nesta tarefa

Referências

- Material didático da disciplina disponível no Teams
- Material didático do prof. Murilo Bresciani de Carvalho, adaptado pela profa. Carmen Maria Costa de Carvalho, disponível no Teams (Material suplementar/Outras referências)