INF1018 - Software Básico (2014.2)
Segundo Trabalho

Leia com atenção o enunciado do trabalho e as instruções para a entrega. Em caso de dúvidas, não invente. Pergunte!

Gerador de Código Dinâmico
O objetivo deste trabalho é desenvolver, em C, uma função geracod que implementa um pequeno gerador de código dinâmico para uma linguagem de programação bastante simples, chamada Minima.
A função geracod deverá ler um arquivo texto contendo o código fonte de uma função escrita em Minima e retornar um ponteiro para uma função (criada pelo gerador) que contém o código que executa a função Minima.

A Linguagem Minima

A linguagem Minima contém apenas três tipos de instruções: atribuição, desvio condicional e retorno.

Uma atribuição tem a seguinte forma:
var := varc1 op varc2
Essa instrução faz com que o valor da expressão varc1 op varc2 seja calculado e atribuído a var. var pode ser uma variável local ou um parâmetro da função e varc1 e varc1 podem ser variáveis locais, parâmetros ou constantes inteiras. op é um dos operadores: + - *
As variáveis locais são da forma vi, sendo o índice i utilizado para identificar a variável (ex. v0, v1, etc...). Da mesma forma, os parâmetros são da forma pi, sendo p0 o primeiro parâmetro, p1 o segundo, e assim sucessivamente. A linguagem permite o uso de no máximo 5 variáveis locais e 5 parâmetros.

Na linguagem Minima as constantes são escritas na forma $i. Por exemplo, $10 representa o valor 10 e $-10 representa o valor -10.

Alguns exemplos de atribuição:

v0 := p0 + p1
v1 := v0 * $100
p0 := p0 - $1
Um desvio condicional tem a seguinte forma:
'ifeq' varc1 varc2 num
Essa instrução faz com que o programa desvie para a instrução na linha de número num se os valores de varc1 e varc2 forem iguais.

Exemplos:

ifeq p0 p1 3
ifeq p0 $0 6
A instrução de retorno tem a forma:
'ret' varc
Essa instrução faz com que a função em execução retorne, com varc como valor de retorno.
Alguns exemplos:

ret $1
ret v0
A sintaxe da linguagem Minima pode ser definida formalmente como abaixo:

func :: cmd '\n' | cmd '\n' func

```
cmd    ::    att | if | ret
att    ::    var ':=' varc op varc
if     ::    'ifeq' varc varc num
ret    ::    'ret' varc
var    ::    'v' digito | 'p' digito
varc   ::    var | '$' snum
op     ::    '+' | '-' | '*'
num    ::    digito | num digito
snum   ::    [-] num
digito ::    0' | '1' | '2' | '3' | '4' | '5' | '6' | '7'| '8' | '9'
```

Exemplos

O exemplo a seguir mostra a implementação da função f(a,b,c) = a * (b + c) na linguagem Minima (os comentários não fazem parte da linguagem).

```
vo := p1 + p2  //1: i = b + c
v0 := p0 * v0  //2: i = a * i
ret v0         //3: retorna i
```

O próximo exemplo implementa uma função que testa se dois números são iguais, retornando 1 caso sejam iguais e 0 caso sejam diferentes:

```
ifeq p0 p1 3   //1: if (a==b) goto 3
ret $0         //2: retorna 0
ret $1         //3: retorna 1
```

O próximo exemplo implementa a função fat(n) em Minima:

```
v0 := $0 + $1  //1: f = 1
ifeq p0 $0 6   //2: if (n == 0) goto 6
v0 := v0 * p0  //3: f = f * n
p0 := p0 - $1  //4: n = n - 1
ifeq $1 $1 2   //5: if (1 == 1) goto 2 -> if (true)
ret v0         //6: retorna f
```

Implementação e Execução

Você deve desenvolver, em C, uma função chamada geracod que leia um arquivo de entrada contendo o código fonte de uma função na linguagem Minima, gere o código de máquina IA32 correspondente, e retorne um ponteiro para função (será um ponteiro para uma área de memória contendo o código gerado).

O arquivo de entrada terá no máximo 50 linhas, com um comando Minima por linha.

O protótipo de geracod é o seguinte:

```
typedef int (*funcp) ();
funcp geracod (FILE *f);
```

O único parâmetro de geracod é o descritor de um arquivo texto já aberto para leitura, de onde deve ser lido o código fonte Minima.

Implementação

A função geracod deve alocar um bloco de memória para escrever o código gerado. O valor de retorno de geracod será um ponteiro para essa área. (Para simplificar, você pode considerar o número máximo de linhas de arquivo na hora de alocar o bloco de memória para código.)

O código gerado por geracod deverá ser um código de máquina IA-32, e não um código fonte assembly. Ou seja, você deverá descobrir o código de máquina que corresponde às instruções de

assembly que implementam cada uma das instruções de nossa linguagem, usando o programa objdump e possivelmente a documentação das instruções da Intel, disponível na página do curso.

Por exemplo, para descobrir o código gerado por movl %eax, %ecx, você pode criar um arquivo meuteste.s contendo apenas essa instrução, traduzi-lo com o gcc (usando a opção -c) para gerar um arquivo objeto meuteste.o, e usar o comando

objdump -d meuteste.o
para ver o código de máquina gerado.
Lembre-se que as instruções assembly ocupam um número variável de bytes na memória.

Não é necessário fazer o tratamento de erros do arquivo de entrada, você pode supor que o código nele estará sempre correto. Vale a pena colocar alguns testes (ver programa exemplo abaixo) só para facilitar a própria depuração do seu código, mas as entradas usadas como testes na correção do trabalho sempre estarão corretas.

Para ler e interpretar cada linha da linguagem Minima, teste se a linha contém cada um dos formatos possíveis. Veja um esboço de código C para fazer a interpretação de código aqui. Lembre-se que você terá que fazer adaptações pois, dentre outros detalhes, essa interpretação não deve ser feita na main!!

IMPORTANTE: Este trabalho não é trivial. Implemente sua solução passo a passo, testando separadamente cada passo implementado! Por exemplo:

Compile um arquivo assembly contendo uma função simples usando:
minhamaquina> gcc -m32 -c code.s
(para apenas compilar e não gerar o executável) e depois veja o código de máquina gerado usando:
minhamaquina> objdump -d code.o
Construa um programa inicial que aloque espaço e coloque esse código "colado" do compilador, bem conhecido, para testar um esqueleto inicial.
Implemente e teste a tradução de uma função Minima que contenha apenas uma instrução de retorno (uma constante, depois um parâmetro de entrada).
Implemente e teste atribuições e operações aritméticas. Implemente uma operação por vez. Experimente usar constantes, parâmetros e variáveis locais como operandos.
Deixe para implementar a instrução de desvio (ifeq) apenas quando todo o resto estiver funcionando!

Testando o gerador de código

Você deve criar um arquivo contendo a função geracod e outro arquivo com uma função main para testá-la.

Sua função main deverá abrir um arquivo texto que contém um "programa fonte" na linguagem Minima (i.e, uma função Minima) e chamar geracod, passando o arquivo aberto como argumento. Em seguida, sua main deverá chamar a função retornada por geracod, passando os parâmetros apropriados, e imprimir o valor de retorno dessa função. Esse retorno é um valor inteiro, que pode ser exibido com código de formação ("%d\n"). A função criada por geracod é a tradução da função Minima lida do arquivo de entrada.

Para testar a chamada de uma função Minima com diferentes parâmetros, sua função main pode receber argumentos passados na linha de comando. Para ter acesso a esses argumentos (representados por strings), a sua função main deve ser declarada como

int main(int argc, char *argv[])

sendo argc o número de argumentos fornecidos na linha de comando e argv um array de ponteiros para strings (os argumentos). Note que o primeiro argumento para main (argv[0]) é sempre o nome do seu executável!. Os parâmetros que deverão ser passados para a função criada por geracod serão o argumento 1 em diante.

Prazo
O trabalho deve ser entregue até meia-noite do dia 24 de novembro.
Será descontado um ponto por dia de atraso.
Entrega
Deverão ser entregues via Moodle dois arquivos:

um arquivo fonte chamado geracod.c , contendo a função geracod (e funções auxiliares, se for o caso).
Esse arquivo não deve conter a função main.
Coloque no início do arquivo, como comentário, os nomes dos integrantes do grupo da seguinte forma:
/* Nome_do_Aluno1 Matricula Turma */
/* Nome_do_Aluno2 Matricula Turma */
um arquivo texto, chamado relatorio.txt, contendo um pequeno relatório.
o relatório deverá indicar o que está funcionando e o que não está funcionando
o relatório deverá mostrar também alguns exemplos de programas da linguagem Minima que você usou para testar o seu trabalho. Mostre tanto os programas Minima compilados e executados com sucesso como os que resultaram em erros (se for o caso).
coloque também no relatório o nome dos integrantes do grupo
Indique na área de texto da tarefa do Moodle o nome dos integrantes do grupo. Apenas uma entrega é necessária (usando o login de um dos integrantes do grupo).
Observações
Os trabalhos devem preferencialmente ser feitos em grupos de dois alunos .
Alguns grupos poderão ser chamados para apresentações orais / demonstrações dos trabalhos entregues.