

Trabalho Prático 2 - Ligador

Lucas Machado

Yuri Niitsuma

Nathalia Campos

Introdução

O trabalho consiste em criar um *linker* para o *montador* do Wombat2 feito no [Trabalho Prático 1 \(Github\)](#).

Assembler/Montador

Modificações

No Montador, tivemos que modificar as classes **SymbolTable** e **OpcodeTable** para prepará-lo para o ligador.

Primeiro adaptamos o **.extern** para guardar o símbolo com o valor de endereço **0** e tipo **EXTERN** para diferenciar dos outros tipos de símbolos usados.

Depois adicionamos o suporte para "*lembrar*" os usos do *labels* (`_label`), *variáveis* (`num`) e *funções* (`.extern`).

Exemplo utilizando o `-v` (verbose) para imprimir a tabela de símbolos.

```
Symbol Table:
varLocal1  0x10  0x2  0x8
varLocal2  0x12  0x6
EXTERN:
funcao1    0x0    0x4
funcao2    0x0    0xa
```

Observe que **varLocal1** é um símbolo local e **funcao1** e **funcao2** são chamadas de funções.

Todos os símbolos tem, no início, o endereço de referência e seguidos pelos endereços de onde ele foi utilizado.

No exemplo utilizado:

```
varLocal1 0x10 0x2 0x8
```

- `0x10` indica o valor a ser substituído na compilação.
- `0x2 0x8` endereços onde o símbolo foi utilizado

Arquivos de saída

Além do arquivo **MIF**, é gravado também o arquivo **SBTL** contendo a tabela de símbolos para o **Linker** utilizá-lo como informação.

```
Size: 0x10
varLocal1 0x10 0x2 0x8
varLocal2 0x12 0x6
EXTERN:
funcao1 0x0 0x4
funcao2 0x0 0xa
```

Utilizando o mesmo formato descrito na seção [Modificações](#) contendo **Size** que é o tamanho ocupado na RAM pelo módulo.

Linker

Implementação

O linker foi implementado do seguinte modo.

- **SymbolTable**: De modo semelhante da classe utilizada no montador adaptado para o Linker, não reutilizei completamente o código pra não dar conflito.
- **Programa**: Classe que contém os arquivos "**MIF**" e "**SBTL**" e contém sua própria tabela de símbolos **locais** e **extern**.
- **Linker**: Classe que gerencia os programas:
 - **output**: Arquivo que vai ser o arquivo ligado e relocado.
 - **mainProg**: O módulo **Mif** principal, será alocado no endereço 0.
 - **modulos**: os módulos adicionais que serão juncionados para o programa **output**.

- **main.cpp**: Utilizado apenas para Instanciar a classe Linker, e utilizar os métodos para criar o arquivo **MIF** concatenado e atualizar os endereços.

O Linker utiliza o padrão de argumentos proposto na especificação, com um bônus de adicionar um modo verbose **"-v"**.

```
linker programa.mif main modulo1 modulo2 ... moduloN [-v]
```

O linker lê todos os módulos (mif, sbtn) atualizando os endereços e concatenando no arquivo programa.mif (por padrão). Neste passo ele atualiza apenas as variáveis/label locais. Ou seja, tem um "escopo" utilizado em cada módulo podendo utilizar o nome da mesma variável em escopos diferentes.

Já os **EXTERN's** são um caso a parte, pois eles são atualizados após o passo acima.

Makefile

Utilizei o [GenericMakefile](#) como template de novo.

- **make**: Compila o linker. Reaproveita os arquivos ".o" que não foram modificados.
- **make clean**: Elimina todos os arquivos compilados.
- **make about**: Executa ./linker apenas para imprimir o *help*.

Ligador Wombat2

```
(C) 2016 Yuri Niitsuma <ignitzhjfk@gmail.com>
      Lucas Machado <lucaspedro.machado@gmail.com>
      Nathalia Campos <nathalia.campos.braga@gmail.com>
```

```
Uso: linker programa.mif main modulo1 modulo2 ... moduloN [-v]
```

- **make ass**: Executa o montador para compilar os módulos dos arquivos .a.
- **make run**: Executa o montador normalmente nos arquivos de testes das **5 opções**.
- **make v**: Executa em modo verbose utilizando os arquivos do **run**. Imprime no terminal todo o processo da linkagem incluindo a tabela de símbolos atualizada após a relocação.

Executar diretamente

O linker pode ser executado diretamente no formato.

```
./linker NOME_ARQUIVO_SAIDA.mif MODULO_MAIN [MODULO1 .. MODULOn] [-v]
```

Lembrando que o arquivo de saída tem extensão "mif" mas os módulos não.

Na compilação é necessária o uso do **c++11** pois utilizo uma funções presentes nesta versão. É compilado sem problemas no **Eufrates**, computador da graduação do **CRC**.

Uso

Se encontra no [README.md](#) contido na pasta do **linker**.

Todos os testes foram *comprovados* na computador da graduação **Eufrates**.

Testes

Segue o pseudo-código (C-like) para indicar o que o programa, depois de linkado, faz:

```

while(true) {
    OP << entrada; // Recebe a operação
    // Entrada dos parâmetros A, B, C
    A << entrada;
    B << entrada;
    C << entrada;
    switch (OP) {
        case 1:
            max(A, B, C);
            break;
        case 2:
            min(A, B, C);
            break;
        case 3:
            sum(A, B, C);
            break;
        case 4:
            prod(A, B, C);
            break;
        case 5:
            media(A, B, C);
            break;
        default:
            exit();
    }
}

```

Seguem screenshots de testes no CPUSIM:

OP é a operação a ser realizada.

- Se $OP = 1$, o programa deverá apresentar o maior dos 3 números,

Wombat2

File Edit Modify Execute Help

Go Step by Instr Step by Micro Backup one Instr Backup one Micro Start Over Fetch sequence: pc->mar Main[mar]->mdr mdr->ir

Data Dec

Registers

Name	Width	Data
buffer1	16	77
buffer2	16	77
ir	16	4094

A

Name	Width	Data
R0	16	77
R1	16	55
R2	16	66

programa.mif X

```

1 DEPTH = 256;
2 WIDTH = 8;
3 ADDRESS RADIX = HEX;
4 DATA RADIX = BIN;
5 CONTENT
6 BEGIN
7
8 00 : 00001111;
9 01 : 11111110;
10 02 : 00001000;
11 03 : 11111110;
12 04 : 10011000;
13 05 : 11111010;
14 06 : 00001000;
15 07 : 11111110;
16 08 : 10011000;
17 09 : 11111110;
18 0A : 00001000;
19 0B : 11111110;
20 0C : 10011000;
21 0D : 11111110;

```

Main

Addr	Data
00	15
01	254
02	8
03	254
04	152
05	250
06	8
07	254
08	152
09	252

Enter Inputs, the first of which must be an Integer: 1
Enter Inputs, the first of which must be an Integer: 1
Enter Inputs, the first of which must be an Integer: 2
Enter Inputs, the first of which must be an Integer: 3
Output: 3
Enter Inputs, the first of which must be an Integer: 1
Enter Inputs, the first of which must be an Integer: 55
Enter Inputs, the first of which must be an Integer: 66
Enter Inputs, the first of which must be an Integer: 77
Output: 77
Enter Inputs, the first of which must be an Integer:

- Se OP = 2, o programa exibirá o menor dos 3,

Wombat2

File Edit Modify Execute Help

Go Step by Instr Step by Micro Backup one Instr Backup one Micro Start Over Fetch sequence: pc->mar Main[mar]->mdr mdr->ir

Data Dec

Registers

Name	Width	Data
buffer1	16	420
buffer2	16	420

A

Name	Width	Data
R0	16	420
R1	16	420

programa.mif X

```

10 02 : 00001000;
11 03 : 11111110;
12 04 : 10011000;
13 05 : 11111010;
14 06 : 00001000;
15 07 : 11111110;
16 08 : 10011000;
17 09 : 11111110;
18 0A : 00001000;
19 0B : 11111110;
20 0C : 10011000;
21 0D : 11111110;
22 0E : 01101001;
23 0F : 00000001;
24 10 : 00100001;
25 11 : 11100000;
26 12 : 01000001;
27 13 : 00101110;

```

Main

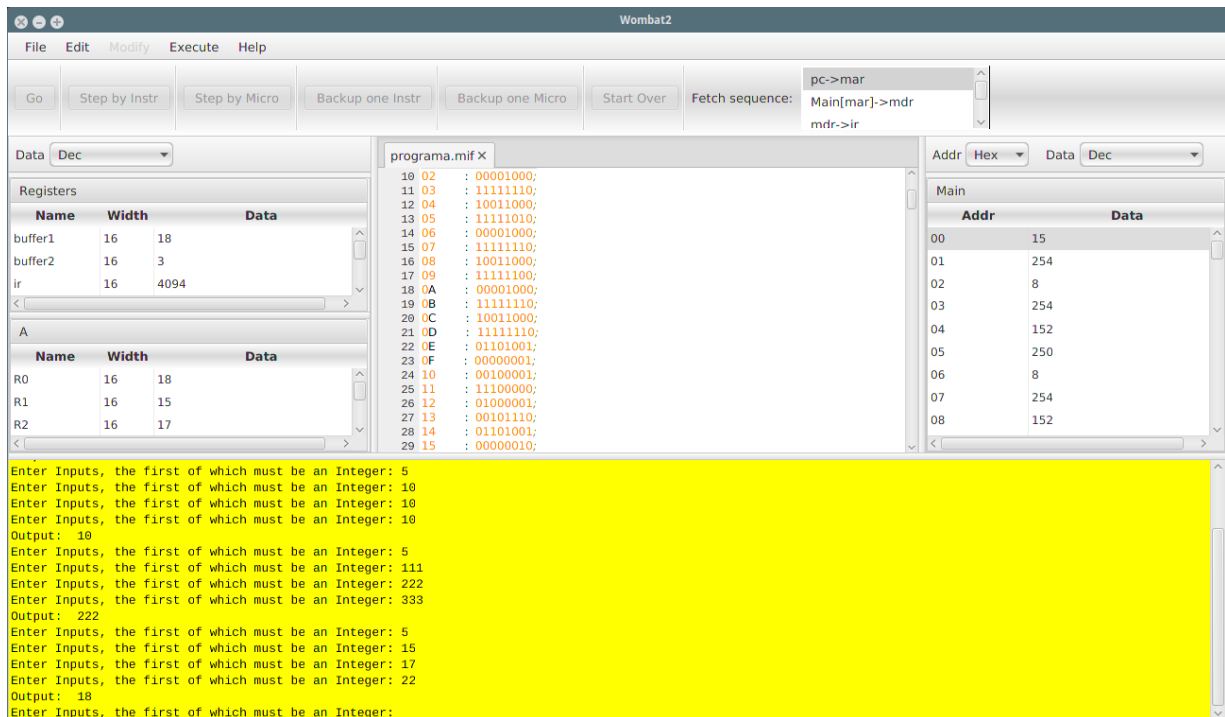
Addr	Data
00	15
01	254
02	8
03	254
04	152
05	250
06	8
07	254

Enter Inputs, the first of which must be an Integer: 2
Enter Inputs, the first of which must be an Integer: 0
Enter Inputs, the first of which must be an Integer: 1
Enter Inputs, the first of which must be an Integer: 2
Output: 0
Enter Inputs, the first of which must be an Integer: 2
Enter Inputs, the first of which must be an Integer: 100
Enter Inputs, the first of which must be an Integer: 200
Enter Inputs, the first of which must be an Integer: 255
Output: 100
Enter Inputs, the first of which must be an Integer: 2
Enter Inputs, the first of which must be an Integer: 420
Enter Inputs, the first of which must be an Integer: 420
Enter Inputs, the first of which must be an Integer: 420
Output: 420
Enter Inputs, the first of which must be an Integer:

- Se $OP = 3$, o programa exibirá a soma dos 3,

- Se $OP = 4$, o programa exibirá o produto dos 3,

- Se $OP = 5$, o programa exibirá a média dos 3,



- Para valores de OP diferentes de 1, 2, 3, 4 ou 5, o programa não fará nada. Nesse caso ele dá um jump pro comando exit

Durante o desenvolvimento do linker utilizei os arquivos **tp2teste2main.a**, **tp2teste2calc.a** e **tp2teste2fact** disponibilizado pra testes no TP1 para usa-los neste TP.

Ideia descartada:

- Tivemos a ideia de dar suporte a pseudo instruções no .extern tal que faria a chamada **call** e **troca de contexto** dos 8 registradores mais o **RA** para dar suporte a recursão, etc. Mas desistimos de inserir pois com o programa teste das 5 opções não caberia nos 255 bytes da memória RAM pois o bloco da pilha (**sp**) encontraria rapidamente com o bloco do programa (semelhante ao **.text** do UNIX).

Isso é tudo corretor. A qual é a boa de hoje é:

- [Melhor jogo da mundial 2016 de League of Legends](#) (Apesar do Samsung ter perdido)
- [Black Mirror](#) (Sensacional)

Referências

- [cplusplus Reference C++11](#)
- Organização Estruturada de Computadores - Andrew S. Tanenbaum (sexta edição)
- CPUSim Help em HTML que vem no pacote