

# Trabalho Prático 1

Fernanda Aparecida R. Silva<sup>1</sup>

<sup>1</sup>Universidade Federal de Minas Gerais (UFMG)  
Belo Horizonte – Minas Gerais – MG – Brazil

{fernandaars}@dcc.ufmg.br

## 1. Introdução

Nesse trabalho foi implementado um sistema de programação composto por um montador (*assembler*), um *linker*, um carregador (*loader*) e um simulador (*runner*). Por fim, também foi criado um conjunto de macros de forma a possibilitar programação em "alto-nível".

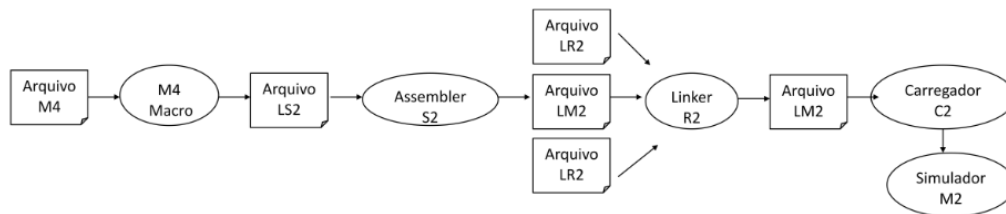


Figura 1. Esquema de Funcionamento do Sistema

## 2. Módulos

Seguem-se abaixo a descrição detalhada de cada um dos módulos implementados e seus respectivos funcionamentos:

### 2.1. M2 - Carregador e Simulador

No módulo M2 foram criados um carregador e um simulador. O objetivo do carregador é carregar o conteúdo de programa na linguagem LM2 para um determinado espaço de memória. Já o simulador é encarregado de, através do conteúdo do programa previamente carregado, executar os comandos determinados no código, modificando variáveis e registradores.

Para utilizar o módulo M2, basta compilá-lo e executá-lo como se segue:

```
make M2
./M2 <arquivo de entrada em LM2> <endereço de carregamento>
```

### 2.2. S2 - Montador

Em S2 foi implementado um montador capaz de fazer a transformação da linguagem de montagem para a linguagem de máquina. O montador traduz as instruções, interpreta as pseudo-instruções, transformando-as ou não em instruções, e faz as primeiras correspondências de endereçamento para os endereços disponíveis dentro do código.

Para utilizar o módulo M2, basta compilá-lo e executá-lo como se segue:

Código	Símbolo	Descrição	Ação
01	LAD M	Carga AC direto	AC <- mem[M + PC]
02	SAD M	Armazena AC direto	mem[M + PC] <- AC
03	ADD M	Soma	AC <- AC + mem[M + PC]
04	SUB M	Subtrai	AC <- AC - mem[M + PC]
05	INP M	Entrada	mem[M + PC] <- inteiro lido
06	OUT M	Saída	Escreve mem[M + PC]
07	JMP M	Desvio	PC <- M + PC
08	JGZ M	Desvio se AC maior que zero	Se AC >0, então PC <- M + PC
09	JLZ M	Desvio se AC menor que zero	Se AC <0, então PC <- M + PC
10	JZE M	Desvio se AC igual a zero	Se AC == 0, então PC <- M + PC
11	HLT	Para a execução	Termina o programa
12	LXD M	Carrega RX direto	RX <- mem[M + PC]
13	SXD M	Armazena RX direto	mem[M + PC] <- RX
14	LAX	Carrega AC indexado	AC <- mem[RX] RX <- RX + 1
15	SAX	Armazena AC indexado	mem[RX] <- AC RX <- RX + 1
16	LCD M	Carrega RC direto	RC <- mem[M + PC]
17	JCC M	Conta e desvia se RC >0	RC <- RC - 1; Se RC >0, então PC <- M + PC
18	CAL M	Chama subprograma	RX <- PC PC <- M + PC
19	RET	Retorna	PC <- RX
20	LAI	Carrega AC indireto	AC <- mem[mem[RX]] RX <- RX + 1
21	SAI	Armazena AC indireto	mem[mem[RX]] <- AC RX <- RX + 1
22	DOB M	Calcula dobro	mem[M + PC] <- 2 · mem[M + PC] AC <- mem[M + PC]
23	MET M	Calcula metade	mem[M + PC] <- mem[M + PC] / 2 AC <- mem[M + PC]
24	JPA M	Desvio se AC par	Se AC for par, então PC = M + PC

**Figura 2. Instruções Reconhecidas na Linguagem LM2**

```
make S2
./S2 <arquivo de entrada em LS2> <arquivo de saída em LM2>
```

### 2.3. R2 - Linker

No módulo R2 foi construído o linker, software usado para ligar os códigos gerados pelo montador, uma vez que um programa pode ser escrito utilizando subprogramas definidos em arquivos separados.

Para utilizar o linker criado, basta compilá-lo e executá-lo como se segue:

```
make R2
./R2 <arquivo de entrada 1 em LM2> ... <arquivo de entrada N em LR2>
```

### 2.4. M4 - Macros

O conjunto de macros criadas, bem como sua "tradução" para a linguagem de montagem encontra-se abaixo:

program	JMP LBL_PROG . z DC 0 . u DC 1
begin	LBL_PROG.
ler	INP 1
escrever	OUT 1
soma	LAD 2 . ADD 3 . SAD 1
sub	LAD 2 . SUB 3 . SAD 1
inc	LAD 1 . ADD u . SAD 1
dec	LAD 1 . SUB u . SAD 1
clr	LAD z . SAD 1
int	1 DC 2
end	HLT
end_program	END

### 3. Conclusão

Concluo, através da execução do trabalho, a grande validade da construção de um sistema de programação, já que para implementá-lo é necessária a compreensão, mesmo que ainda superficial, de todo o processo pelos quais os programas passam em um computador real, da linguagem de alto-nível até a linguagem de máquina. Através do mini-sistema desenvolvido, foi possível emular conceitos complexos com processos mais simples do que os necessários nos softwares convencionais. Assim, o aprendizado torna-se mais fácil sem no entanto perder a verossimilhança com a vida real. Além disso, com o trabalho foi possível também experienciar situações e erros que podem ocorrer no decorrer da nossa vida profissional de forma a me preparar para ele.