

# Trabalho Prático 1

Semar Augusto da Cunha Mello Martins

## 1. Introdução:

Ao longo do curso são ensinados os conceitos de máquina virtual, simulador, carregador, montador, expensor de macros e ligador. O objetivo do trabalho é fazer com que implementemos um mini-sistema que possa executar programas escritos em uma linguagem definida pelo professor com o uso de todos os conceitos citados a cima.

- . A Máquina virtual é um programa que simula um computador. Ele é programado para se comportar exatamente como uma máquina, mas sem nenhum componente físico.
- . O Simulador é a máquina virtual propriamente dita. É um programa em C que recebe instruções de máquina e as executa.
- . O Carregador é uma função parte do Simulador que carrega um programa em M2 para a memória do Simulador.
- . O Montador traduz um programa escrito em assembly para linguagem de máquina.
- . O Expensor de Máquinas é
- . O Ligador é um programa que recebe os códigos já montados pelo montador e os une com outros arquivos como informações de relocação de memória e gera um novo arquivo completo em linguagem de máquina.

## 2. Solução Proposta:

Para o Simulador foi criado um struct que possui uma memória com tamanho fixo em 256 inteiros, inteiros que delimitam onde o uso da memória começa e termina, 4 registradores, PC, RX, RC e AC e um flag que determina se a última operação resultou em um 0 ou não.

O Carregador é apenas uma função que recebe onde deve começar a colocar o programa na memória, lê o arquivo e o coloca lá.

Para o Montador foram criados uma tabela de variáveis, uma tabela de labels e uma tabela de instruções.

O Expensor de Máquinas e o Ligador não foram implementados no trabalho.

## 3. Funções:

Simulador:

```
initInterpreter; // inicializa o simulador, alocando memória e inicializando os  
valores que precisam ser inicializados  
readFile; // é o carregador  
getInstruction; // pega a próxima instrução a ser executada e anda com o PC  
exec; // é onde o programa realmente começa a executar o programa dado  
como entrada.  
decode; // é a função que pega um opcode e um deslocamento, identifica o  
que eles deveriam fazer e faz.
```

Montador:

*initMounter; // inicializa o montador, alocando memória e inicializando os valores que precisam ser inicializados*

*createSymbolTable; // faz uma primeira leitura no arquivo e preenche as tabelas de variável e label.*

*runProgram; // faz a transformação de uma linguagem M2 para linguagem de máquina.*

### **3. Implementação:**

*Para a implementação do simulador foram tomadas algumas decisões como criar uma estrutura para a máquina virtual, dividir o código em tokens individuais (o programa só pega o primeiro comando da linha e decide se pega um segundo ou não a partir do número que foi lido).*

*A decisão de criar uma estrutura de dados foi feita colocando todos os registradores necessários nela e uma memória de 256 inteiros.*

*Para a implementação do montador foram criadas várias estruturas, uma que lê a linha inteiro de um programa s2 e guarda as informações que estão nela. uma estrutura que determina o nome a a posição em que uma label está e uma estrutura da mesma forma para variáveis. Por fim, há uma estrutura final chamada mounter com uma tabela de comandos, uma tabela de variáveis e uma tabela de labels.*

### **4. Compilação e Execução:**

*Para compilar os programas é necessário apenas entrar na pasta src e escrever o comando “make” que compilará tanto o montador quanto o simulador.*

*Para a execução do programa, o makefile já está pronto para executar o programa, possui uma variável chamada “sim” e outra variável chamada “mont”, onde se colocam os argumentos para execução e o comando “make sim” e “make mont” executam o simulador e o montador respectivamente.*

*A estrutura de execução do simulador é: ./M2 <nome do programa .m2 a ser executado> <local da memória onde o programa será colocado> < <nome do arquivo de input do programa a ser executado> (exemplo: ./M2 restoDivisao.m2 10 < input)*

*A estrutura de execução do montador é: ./S2 <nome do programa em linguagem s2 ou ls2> <nome do arquivo .m2 de saída>*

### **5. Conclusão:**

O trabalho prático 1 foi um trabalho extremamente difícil de ser implementado e o prazo inicialmente curto me fez desistir da implementação completa, seria possível implementar mais do trabalho com os aumentos de prazos mas as provas de outras disciplinas me impediram.

Todos os testes executados funcionaram propriamente e acredito que a parte implementada foi feita como pedido.