

Trabalho Prático 2 – Ligador

Este documento descreve o trabalho prático que será usado para treinar e fixar os conceitos aprendidos em sala de aula na disciplina de Compiladores, relacionados ao processo de montagem de um programa. A segunda parte do trabalho consiste em implementar um editor de ligação para uma máquina previamente projetada.

CONSIDERAÇÕES GERAIS

- O trabalho deverá ser implementado obrigatoriamente na linguagem C/C++;
- Deverá ser entregue exclusivamente o código fonte com os arquivos de dados necessários para a execução e um arquivo Makefile que permita a compilação do programa nas máquinas UNIX do departamento;
- Além disso, deverá ser entregue uma pequena documentação contendo todas as decisões de projeto que foram tomadas durante a implementação, sobre aspectos não contemplados na especificação, assim como uma justificativa para essas decisões;
- A ênfase do trabalho está no funcionamento do sistema e não em aspectos de programação ou interface com o usuário. Assim, não deverá haver tratamento de erros no programa de entrada;
- A entrega do trabalho deverá ser realizada por meio do Microsoft Teams, na tarefa criada especificamente para tal;
- **ATENÇÃO:** trabalhos que não seguem esse padrão serão penalizados.

ESPECIFICAÇÃO DO EDITOR DE LIGAÇÃO

Os principais objetivos do Editor de Ligação a ser projetado e implementado neste trabalho são:

- Permitir relocação de programas: endereço de carga de programa deve ser definido somente após a tradução;
- Permitir tradução separada: os módulos de um programa são montados separadamente e depois combinados para formar um único programa objeto.

Algumas informações importantes sobre a implementação são:

- Deverá ser implementado o editor de ligação que combine os diversos sub-programas que foram montados independentemente. Para determinar o ponto de início do programa, considere que todo programa terá um label “main” em um dos módulos;
- Para a implementação do editor de ligação, informação adicional precisa ser gerada pelo montador do **Trabalho Prático 1**. Além de não gerar erros no segundo passo da montagem, devido a símbolos desconhecidos, o arquivo gerado deve conter a tabela de símbolos do programa. O programa gerado pelo montador, portanto, não é necessariamente executável, mas um formato

que servirá de entrada para o editor de ligação, que deve realizar as **3 tarefas: alocação, ligação e relocação**, produzindo, assim, a partir de 1 ou mais arquivos gerados pelo montador, 1 programa executável único, no formato que possa ser carregado e executado na máquina virtual (ver descrição do Trabalho Prático 1).

Para testar o novo montador e o editor de ligação, utilize os dois módulos a seguir (versão modificado do exemplo visto no Trabalho Prático 1):

```
main: READ R0
      STORE R0 a
      CALL add100
      WRITE R0
      HALT
const100: WORD 100
a: WORD -1
END
```

```
add100: LOAD R0 a
        LOAD R1 const100
        ADD R0 R1
        RET
END
```

Algumas observações importantes:

- O montador deverá receber como linha de comando o nome do arquivo texto contendo o programa em assembly e deverá gerar, **na saída padrão**, o arquivo no formato aceito pelo editor de ligação. Esse formato fica a sua escolha;
- O editor de ligação deverá receber como linha de comando os nomes de um ou mais arquivos no formato de saída do montador e deverá gerar, **na saída padrão**, o arquivo executável no formato aceito pela MV;
- Deve ser submetido, junto ao código fonte, um arquivo Makefile, de forma que seja possível compilar o montador e o editor de ligação com o comando “make” (um Makefile para cada, pastas diferentes);
- Ao compilar o montador e o ligador, os arquivos executáveis **devem** ser criados em uma pasta chamada “bin” (nos diretórios em que estão os respectivos arquivos Makefile);
- O executável do montador **deve** se chamar “montador”;
- O executável do montador **deve** se chamar “ligador”;
- É importante que essa padronização seja seguida, em virtude da automatização na correção dos trabalhos.

SOBRE A DOCUMENTAÇÃO ANEXA

Juntamente com o trabalho, cada aluno deverá submeter um documento com informações relevantes sobre seu trabalho, a dizer:

- Deve conter as decisões importantes tomadas nas definições do projeto que por ventura não estejam contempladas na especificação do trabalho;
- Não é importante incluir listagem de código fonte nessa descrição;
- Deve conter elementos que comprovem que o programa foi testado com alguma profundidade.