

JOÃO JOSÉ NETO

"ASPECTOS DO PROJETO DE SOFTWARE DE UM MINICOMPUTADOR"

"Dissertação de Mestrado" apresentada
à Escola Politécnica da Universidade
de São Paulo, para obtenção do título
de Mestre em Engenharia.

Área de Concentração - Engenharia de
Eletricidade.

Orientador: Prof. Dr. Antonio Marcos de Aguirre Massola

São Paulo

-1975-

Ar Prof. Divina
oferece um exemplar
do meu trabalho de
mestrado, com meus agrade-
cimentos 10/11/75

JOÃO JOSÉ NETO

"ASPECTOS DO PROJETO DE SOFTWARE DE UM MINICOMPUTADOR"

"Dissertação de Mestrado" apresentada
à Escola Politécnica da Universidade
de São Paulo, para obtenção do título
de Mestre em Engenharia.

Área de Concentração - Engenharia de
Eletricidade.

Orientador: Prof. Dr. Antonio Marcos de Aguirra Massola

São Paulo

-1975-

A meus pais e irmã.

A G R A D E C I M E N T O S

Ao Prof. Dr. Antonio Marcos de Aguirra Massola, pela dedicação com que acompanhou, orientou, apoiou e incentivou este trabalho.

Aos Professores Doutores James Gregory Rudolph, Antonio Hêlio Guerra Vieira e Tamio Shimizu pelas idéias apresentadas.

Aos Engs. Benício José de Souza, Ting Kong Sen e Wanner Monteiro Pinheiro, e engenheirandos Luiz Sanches Filho, Mário Tachibana e Charlie Lin, pela participação efetiva no projeto e no aprimoramento dos programas desenvolvidos.

Aos Engs. Laércio Antonio Marzagão, Antonio Marcos de Aguirra Massola e Benício José de Souza, à Enga. Selma Shin Shimizu Melnikoff, à Profa. Selenê Cavalcanti Ferrari e aos meus familiares, pelos inúmeros diálogos e constante apoio e estímulo, decisivos para a concretização deste trabalho.

À srta. Sonia Regina Izarelli pelos serviços de datilografia e desenho.

A Valdecir Finco pelo serviços de impressão.

A todos, enfim, que de uma ou outra forma contribuíram para a concepção ou desenvolvimento deste trabalho, particularmente à equipe de estagiários do Laboratório de Sistemas Digitais, sem cuja valiosa participação não teria sido possível a realização deste trabalho.

R E S U M O

O presente trabalho descreve os métodos utilizados no desenvolvimento de alguns dos módulos do "software" básico do Patinho Feio, o primeiro minicomputador desenvolvido no Laboratório de Sistemas Digitais do Departamento de Engenharia de Eletricidade da Escola Politécnica da Universidade de São Paulo.

Para cada módulo apresentado, são discutidos os seus objetivos, sendo, quando conveniente, apresentados alguns dos problemas enfrentados durante seu desenvolvimento específico ou então problemas mais gerais, referentes ao projeto de programas semelhantes ao mesmo. Sempre que possível, são apresentadas alternativas de solução dos referidos problemas, descrevendo-se finalmente alguns detalhes de um exemplo de implementação.

No Capítulo 2 são discutidos mais profundamente os problemas enfrentados durante o projeto e a implementação de um montador, desde a sua concepção até a sua implantação definitiva, enfatizando-se a metodologia empregada na implementação e os critérios adotados nas decisões mais importantes.

No Capítulo 3 é descrito um simulador-interpretador, implementado em outro computador com a intenção de auxiliar o desenvolvimento do "software" básico do Patinho Feio. Algumas técnicas de simulação são discutidas neste Capítulo, ao lado dos algoritmos utilizados pelo interpretador.

Nos demais Capítulos, são descritos mais alguns programas do "software" básico desenvolvidos para o Patinho Feio, tais como um desmontador, um programa para auxiliar a depuração de outros programas e um editor simbólico.

Nos apêndices, são apresentados tópicos julgados convenientes para a complementação de algumas idéias, bem como alguns exemplos de utilização de programas apresentados neste trabalho.

A B S T R A C T

The present work describes the methods that had been employed during the development of some of the basic software modules of the "Patinho Feio", the first minicomputer of the "Laboratório de Sistemas Digitais da Escola Politécnica da Universidade de São Paulo".

For each module, after a brief discussion of their objectives, some of the main problems which had to be solved during their development are presented, as well as more general ones, which arrive when designing programs of the same class. Whenever possible, alternative solutions of these problems are presented, and, at last, an example of implementation is described.

Chapter 2 discusses in detail the problems arrived during the design and implementation of an assembler, from the phase of its conception until that of its final installation, emphasizing the methods employed in the implementation and the criteria which were used when the most important decisions had to be taken.

Chapter 3 describes a simulator-interpreter, which had been implemented in an auxiliary computer, and was intended to help the development of the basic software of the "Patinho Feio". This chapter discusses also some simulation techniques and the algorithms employed in the interpreter.

The remaining chapters describe some additional programs of the basic software developed for the "Patinho Feio", like a disassembler, a debugging routine and a symbolic editor.

The appendices present some subjects considered helpful for completing some ideas, and some execution examples of the programs presented in this work.

I N D I C E

1. INTRODUÇÃO

- 1.1 - Objetivos
- 1.2 - Generalidades
- 1.3 - Observações

2. O MONTADOR

- 2.1 - O conjunto de instruções e a linguagem do montador
 - 2.1.1 - O conjunto de instruções
 - 2.1.2 - Programação em linguagem de máquina
- 2.2 - A conveniência de um montador
 - 2.2.1 - Rotinas auxiliares para elaboração do montador
- 2.3 - Definição das características gerais do montador
- 2.4 - Definição das características externas do montador
 - 2.4.1 - Características do Montador Absoluto
 - 2.4.2 - Características do Montador Relocável
 - 2.4.3 - A sintaxe da Linguagem de Entrada
 - 2.4.4 - Características do Código Objeto gerado pelo Montador Absoluto
 - 2.4.5 - Características do Código Objeto Gerado pelo Montador Relocável
- 2.5 - Definição das Características Internas do Montador
 - 2.5.1 - A representação interna dos rótulos
 - 2.5.2 - A organização da tabela de símbolos
 - 2.5.3 - A manipulação da tabela de símbolos
 - 2.5.4 - A organização e a manipulação de tabela dos mnemônicos
 - 2.5.5 - A representação interna das Constantes
 - 2.5.6 - As pseudos instruções
 - 2.5.6.1 - A pseudo ORG
 - 2.5.6.2 - As pseudos NOME, SUBR, SEGM
 - 2.5.6.3 - A pseudo DEFC
 - 2.5.6.4 - A pseudo COM

- 2.5.6.5 - A pseudo BLOC
- 2.5.6.6 - A pseudo EQU
- 2.5.6.7 - O Controle BLT e D
- 2.5.6.8 - As pseudos DEFE e DEFI
- 2.5.6.9 - A pseudo FIM
- 2.6 - O programa principal
 - 2.6.1 - O primeiro passo
 - 2.6.2 - O segundo passo
 - 2.6.3 - Observações sobre a Ampliação dos Recursos do Montador
 - 2.6.4 - Críticas
- 3. UM SIMULADOR-INTERPRETADOR PARA A LINGUAGEM DE MÁQUINA DO PATINHO FEIO
 - 3.1 - Definição das Especificações do programa de simulação
 - 3.2 - O interpretador das instruções
 - 3.3 - A simulação do sistema de entrada e saída
 - 3.3.1 - O modelo do Sistema de Entrada e Saída
 - 3.3.2 - A interação entre o interpretador de instruções e o simulador de entrada e saída
 - 3.4 - O programa controlador
 - 3.4.1 - A fase de Console
 - 3.4.2 - A fase de Execução
 - 3.5 - Comentários, Críticas e Sugestões
- 4. UM DESMONTADOR PARA A LINGUAGEM DE MÁQUINA DO PATINHO FEIO
 - 4.1 - Especificações do Desmontador
 - 4.2 - A Lógica do Desmontador
 - 4.2.1 - O primeiro passo do desmontador
 - 4.2.2 - O segundo passo do desmontador
 - 4.3 - Alguns detalhes de Implementação
 - 4.3.1 - Desmontador Absoluto
 - 4.3.2 - Desmontador Relocável
 - 4.4 - Conclusões

5. A ROTINA DE DEPURAÇÃO DE PROGRAMAS

- 5.1 - A fase de depuração de um programa
- 5.2 - A filosofia de rotina de depuração
- 5.3 - Os problemas enfrentados
- 5.4 - Exemplo de Implementação
 - 5.4.1 - Rotina de Entrada de Dados
 - 5.4.2 - Rotina de Interpretação e Execução de Linguagem de Máquina
 - 5.4.3 - Rotina de Relatório
 - 5.4.4 - Comentários e Observações

6. O EDITOR SIMBÓLICO

- 6.1 - Introdução
- 6.2 - A filosofia do Editor
 - 6.2.1 - Definição do Conjunto de instruções do editor
- 6.3 - A lógica do Editor
- 6.4 - Um exemplo de Implementação
- 6.5 - Comentários sobre alguns detalhes de Implementação

APÊNDICE 1 - O CONJUNTO DAS INSTRUÇÕES DE MÁQUINA DO PATINHO FEIO

- A1.1 - Grupo 1 - Instruções de Referência à Memória
- A1.2 - Grupo 2 - Instruções de Endereçamento Imediato
- A1.3 - Grupo 3 - Instruções de Deslocamentos e Giros
- A1.4 - Grupo 4 - Instruções de Entrada e Saída
- A1.5 - Grupo 5 - Instruções Curtas com operando
- A1.6 - Grupo 6 - Instruções Curtas sem operando

APÊNDICE 2 - ROTINAS AUXILIARES UTILIZADAS NA CONSTRUÇÃO DO "SOFTWARE" BÁSICO

- A2.1 - A rotina de geração de fita perfurada carregável ("dumper")
- A2.2 - O carregador de Fitas Absolutas

- A2.3 - A Rotina de Listagem do Conteúdo da Memória
- A2.4 - A Rotina de Carregamento de Memória a partir de dados em hexadecimal
- A2.5 - Comentários

APÊNDICE 3 - FORMATOS DE FITA OBJETO

- A3.1 - Formato de fita Objeto Absoluta
- A3.2 - Formato de fita Objeto Relocável

APÊNDICE 4 - ALGUNS EXEMPLOS DE UTILIZAÇÃO DOS PROGRAMAS DESENVOLVIDOS

1. INTRODUÇÃO

1.1 - Objetivos

No presente trabalho expõe-se os métodos utilizados no projeto e construção de algumas das peças do "software" básico do Patinho Feio, o primeiro minicomputador desenvolvido no Laboratório de Sistemas Digitais (LSD) do Departamento de Engenharia de Eletricidade da Escola Politécnica da Universidade de São Paulo. Descreve-se as diversas fases do projeto, analisando detalhadamente os principais problemas que cada uma delas apresenta. Esta análise é feita em nível geral, quando se tratar de problemas que devam ser enfrentados independentemente da máquina, ou então em nível particular, em caso contrário. Esta descrição destina-se principalmente à orientação de futuros projetos semelhantes.

A análise dos problemas dependentes da máquina, bem como a dos detalhes de implementação dos programas descritos, restringe-se ao caso particular dos exemplos implementados, devendo-se levar em conta a configuração do sistema para o qual foram desenvolvidos. Assim, algumas das soluções apresentadas em tais exemplos podem não ser vantajosas em sistemas que não tenham as mesmas características.

São apresentadas alternativas para as soluções de alguns dos problemas enfrentados, fazendo-se uma análise comparativa das mesmas. Procurou-se, para isto, ressaltar vantagens e desvantagens de cada uma das alternativas mencionadas, terminando-se por optar por uma delas e, sempre que possível, justificando-se a opção adotada na implementação do exemplo, com uma argumentação baseada nas características e limitações do computador utilizado.

Além disto, é descrito aqui o funcionamento dos programas implementados, especialmente com a finalidade de documentar as linhas de raciocínio e a filosofia de projeto e de implementação adotadas para facilitar futuras alterações de tais programas.

1.2 - Generalidades (ref. 11, 14)

Nos próximos Capítulos estão desenvolvidos, dentro das linhas apresentadas em 1.1, os seguintes módulos do "software" :

- um montador ("assembler"), cuja finalidade é a de dispensar o programador da tarefa de programação em linguagem de máquina, permitindo que os programas sejam escritos em linguagem simbólica mnemônica. Uma das versões implementadas dá, além disto, a possibilidade de modularização dos programas, introduzindo a facilidade de se utilizar programas escritos e traduzidos independentemente sem a necessidade de nova tradução a cada utilização dos programas parciais;

- uma rotina de depuração, cuja finalidade é a de facilitar a pesquisa e correção de erros existentes em programas ainda não depurados;

- um desmontador ("disassembler"), cuja finalidade é a de gerar programas escritos na linguagem do montador, a partir de uma fita binária contendo o programa correspondente em linguagem de máquina;

- uma rotina de edição de arquivos ASCII, cuja finalidade é a de mecanizar a correção, a reprodução, a modificação e a listagem de arquivos ASCII, como por exemplo, programas perfurados, em linguagem simbólica, em fita de papel ("programas fonte").

Os quatro programas citados acima foram desenvolvidos para o Patinho Feio estando atualmente à disposição para execução no mesmo.

Alguns programas de utilidade foram implementados no computador Hewlett-Packard 2116-B, com a finalidade de facilitar e de tornar mais versátil a utilização do "software" do Patinho Feio.

Escolheu-se o computador HP-2116-B principalmente pela sua compatibilidade de entrada e saída com a do Patinho Feio (fita de papel). Os programas desenvolvidos para o HP-2116-B não foram implementados diretamente no Patinho Feio devido a algumas restrições atualmente existentes no mesmo, como por exemplo sua pequena capacidade de memória, a inexistência de memória de massa e a baixa velocidade dos periféricos disponíveis. Entretanto, poderão alguns desses programas vir a ser implantados no Patinho Feio assim que estiverem disponíveis no mesmo os periféricos necessários. Os programas aqui apresentados são os seguintes:

- um simulador em nível de registradores, que funciona como um interpretador da linguagem de máquina do Patinho Feio e que também permite a utilização dos periféricos rápidos do HP-2116-B em substituição aos disponíveis no Patinho Feio, bem como o acompanhamento do programa em execução por meio de rastreamento ("traces"), descarregamentos ("dumps") de memória, monitoramento dos processos de entrada e saída, etc. Naturalmente a utilização deste simulador-interpretador é às vezes mais conveniente que a do próprio Patinho Feio, como por exemplo, quando o programa for limitado em velocidade pela entrada e saída (caso de rastreamento e de listagens extensas com pouco processamento);

- um montador ("cross-assembler") equivalente ao desenvolvido para o Patinho Feio, que incorpora uma opção de geração da tabela de referências cruzadas ("cross-reference symbol table") e ordenação alfabética da tabela de símbolos, permitindo utilização total dos recursos do sistema operacional DOS ("disc operating system") do computador HP-2116-B. A tabela de referências cruzadas tem como finalidade facilitar a documentação e mesmo a depuração de programas extensos, gerando, a partir da linguagem simbólica, uma tabela em ordem alfabética, de todos os rótulos ("labels", isto é, nomes simbólicos dados aos endereços das instruções ou dos dados do programa), ao lado de cada qual são listados o número da linha onde o mesmo foi definido e o conjunto de todas as linhas onde foi referenciado. Indica também os rótulos indefinidos, os não utilizados, e o endereço de memória associado a cada rótulo.

1.3 - Observações

- Os dois computadores do LSD em que se efetuou a elaboração dos programas descritos neste trabalho, foram utilizados - com a seguinte configuração:

PATINHO FEIO:

- 1 Unidade Central de Processamento, com memória de núcleos de ferrite com 4K palavras de 8 bits (1K = 1024).
- 1 Leitora Ótica de Fita de Papel HP-2737-A, 300 caracteres por

segundo (máximo).

- 2 Terminais Teletype ASR33 com leitora e perfuradora de fita de papel, entrada por teclado e saída impressa, 17 caracteres por segundo (perfuração opcional).

HEWLETT-PACKARD 2116-B:

- 1 Unidade Central de Processamento, com memória de núcleos de ferrite com 16K palavras de 16 bits.
- 1 Unidade de Disco Magnético HP-2770-A, com capacidade de 3 Megabits, organizados em 64 trilhas, de 92 setores cada, tendo cada setor 64 palavras de 17 bits.
- 1 Impressora HP-2767-A de 88 colunas por linha, 1118 linhas por minuto (máximo).
- 3 Unidades de Fita Magnética HP-7970-A, velocidade máxima 37.5 ips, densidade 888 bpi.
- 1 Console Teletype ASR35, de 178 caracteres por minuto.
- 1 Leitora Ótica de Fita de Papel HP-2748-B, de 588 caracteres por segundo (máximo).
- 1 Perfuradora de Fita de Papel HP-2895-B, de 75 caracteres por segundo.
- 1 Leitora de Cartões de marcas Óticas HP-2761-A, de 288 cartões por minuto.

- Os programas desenvolvidos para o Patinho Feio foram inicialmente projetados para a configuração mínima acima descrita. Entretanto, com a inclusão de novos periféricos, é conveniente que se configure os programas existentes, adaptando-os aos novos equipamentos disponíveis, o que pode tornar mais rápida e eficiente a execução de tais programas. Assim, todas as vezes que um

novos periféricos é incorporado ao sistema, faz-se a adaptação dos programas às novas condições, permitindo que sejam utilizados os recursos implantados.

Atualmente, dispõe-se dos seguintes periféricos adicionais, já incorporados ao sistema pela equipe do LSD:

- 1 Impressora de linha de 80 colunas HP-2767-A (1110 linhas/minuto).
- 1 Terminal DECWRITER DEC LA 398, de 80 colunas, 30 caracteres por segundo (máximo).
- 1 Perfuradora Rápida de Fita de Papel HP-2735-A, de 115 caracteres por segundo.

Os programas menos extensos permitem a inclusão de uma "seção de configuração", parte do programa que o adapta para a utilização dos periféricos desejados. Os mais extensos, no entanto, não dispõem de memória livre suficiente que possa alojar a seção de configuração. Para estes programas, a configuração pode ser feita manualmente modificando-se certas posições de memória, ou, então montando-se novamente o programa com as devidas alterações.

- Em todos os programas aqui descritos, procurou-se orientar o projeto de tal modo que o programa resultante apresentasse o maior número possível de recursos por unidade de área de memória ocupada. É evidente que, na maioria das vezes, uma tentativa de adicionar mais recursos ao programa sem aumentar a área ocupada pelo mesmo pode trazer problemas no desempenho do programa, tais como queda de velocidade, quebra de estrutura, etc. Em alguns destes casos, como será visto, optou-se por soluções de compromisso, e, em outros, por soluções que dessem ao programa maiores recursos de utilização.

No presente caso, tem-se os seguintes fatores a considerar :

- a) baixa capacidade de memória (4K palavras de 8 bits)

b) baixa velocidade de saída dos dados (teletype: 10 pa-
lavras por segundo)

c) deficiência do conjunto de instruções no que se refe-
re a:

- manipulação de dados com mais de uma palavra (p/ex,
dados em precisão dupla).
- comparação de dados
- extração e inserção de campos dentro de uma palavra
(inclusive mascaramentos)
- operações aritméticas

Analisando-se os fatores a) e b), levando-se em conta
observações realizadas nos programas aqui descritos, chegou-se às
seguintes conclusões:

- O tempo de processamento é desprezível em relação
ao de entrada e saída. Isto permite concluir que mes-
mo se o programa for algumas vezes mais lento, o
tempo total de execução será praticamente o mesmo.
- Rotinas otimizadas em relação ao tempo de processa-
mento ocupariam, em certos casos, até cerca de 40%
a mais de memória em relação a rotinas equivalentes,
otimizadas em relação ao número de palavras ocupa-
das pela mesma na memória. Levando-se em conta o
fator a), percebe-se que, quando a dimensão do pro-
grama for comparável à dimensão da memória disponí-
vel, pode-se chegar até a uma condição de inviabili-
dade de implementação do mesmo se se insistir em
utilizar rotinas otimizadas em velocidade. Deve-se
observar que tais rotinas sejam, em média, poucas
vezes mais rápidas, o que já foi dito ser pratica-
mente irrelevante no caso.
- Muitas vezes, na tentativa de se reduzir o espaço

ocupado na memória para executar uma certa tarefa, pode ser encontrada uma organização eficiente dos dados que permita a utilização de rotinas rápidas com baixo gasto de memória. Para isso, deve ser tirado o máximo proveito de características particulares da arquitetura da máquina.

- Deve-se levar sempre em conta que este trabalho é dirigido para um computador pequeno e restrito. Assim sendo, algumas das argumentações aqui apresentadas podem não ser válidas para máquinas maiores, e algumas considerações, decisivas neste contexto, podem tornar-se totalmente irrelevantes se no lugar do Patinho Feio for colocado um computador de parte maior.

