

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-8808

**Um Sistema Simples para Documentação
Semi-Automática de Programas**

V. W. SETZER

SETEMBRO 1988

1. INTRODUÇÃO.

A documentação de programas é um dos aspectos cruciais do desenvolvimento de sistemas computacionais em qualquer ambiente. Desde o desenvolvimento de "software" básico, até o de "software aplicativo" para processamentos administrativos, a atividade de programação usando alguma linguagem de programação requer um esforço específico na documentação de sistemas e programas. Esse fato independe da linguagem: essa necessidade existe desde a programação em linguagem de montagem ("assembler") até o uso de linguagens denominadas de "4ª geração".

Temos verificado continuamente que as documentações de sistemas e programas em geral simplesmente inexistem ou são inadequadas e desatualizadas. As principais consequências desses fatos são:

- C1. A manutenção de programas é muito cara e demorada.
- C2. A fase de testes de programas demanda um tempo exagerado e sobrecarrega os analistas e programadores.
- C3. Não há segurança e confiança de que os programas estão corretos.
- C4. O autor de um programa torna-se o "pai" do mesmo, pois ninguém mais consegue compreendê-lo e mantê-lo.
- C5. Não há meios para se verificar a documentação, isto é, se ela realmente reflete a estrutura do programa.
- C6. A gerência não consegue saber se um programa está bem documentado.
- C7. O projeto, a programação e a documentação devem ser feitos pelas mesmas pessoas.

Neste artigo apresentamos um sistema extremamente simples de documentação de programas através de comentários feitos na sintaxe (de comentários) da própria linguagem de programação do programa. Um programa também simples (apresentado aqui como exemplo de documentação em sua versão em Pascal para o VAX) é empregado para gerar automaticamente três níveis de documentação. Delineamos também um programa para gerar automaticamente a documentação das alterações efetuadas em um programa, feitas em um editor de textos qualquer. Veremos como esse sistema também pode forçar metodologias "top-down" de programação (refinamentos sucessivos), como permite introduzir e controlar padrões de documentação e como diminui as consequências apresentadas acima.

2. NÍVEIS DE DOCUMENTAÇÃO.

Em nosso sistema empregamos três níveis de documentação; ele pode ser facilmente estendido para conter mais níveis, se for conveniente. Os níveis contêm o seguinte:

Nível 1: Informações gerais sobre os programas e procedimentos.

Nível 2: Toda a documentação de nível 1, mais estruturas conceituais de dados e de programas e rotinas.

Nível 3: Todas as documentações de níveis 1 e 2, mais o programa propriamente dito, com documentação de seus detalhes.

Note-se o encaixamento dos conteúdos de cada nível em relação aos níveis anteriores.

3. ESPECIFICAÇÕES DOS NÍVEIS.

A documentação de nível 1 é inserida no programa sob forma de comentários ocupando linhas completas (isto é, não deve haver nada, nessas linhas, além desses comentários), que denominaremos de comentários de nível 1. Eles são identificados colocando-se o dígito '1' após o símbolo de início de comentário da linguagem, que deve ser inserido na posição mais à esquerda da linha (coluna 1), como por exemplo '(*1', '{1' ou '*1' (dependendo da linguagem). O símbolo de fim de comentário deve ser colocado nas últimas posições da linha.

A documentação de nível 2, como vimos, contém a de nível 1, produzida pelos comentários de nível 1, e mais os comentários de nível 2. Estes são linhas de comentários de formato idêntico ao dos comentários de nível 1, excetuando-se o dígito '2' colocado no lugar do '1'.

A documentação de nível 3 contém as duas anteriores, e mais todos os comandos do programa e comentários de nível 3. Estes são inseridos no programa em qualquer posição das linhas, usando-se simplesmente os símbolos de comentários da linguagem empregada. Evidentemente, não devem confundir-se com os comentários de nível 1 e 2; portanto, não devem conter os dígitos '1' ou '2' imediatamente após o símbolo de início de comentário.

4. LISTADOR DA DOCUMENTAÇÃO.

O "listador" da documentação é um programa que solicita do usuário a informação do nível de documentação desejado, e em seguida varre sequencialmente o arquivo com o programa, produzindo uma listagem contendo o seguinte:

- a) Cabeçalho em cada página, contendo o nome do programa, o nível de documentação, a data e a hora da listagem e o número da página.
- b) Se o nível desejado é 3, a listagem conterá todas as linhas do programa, incluindo todos os comentários, precedida cada uma por um número de linha. Esse número é o número de ordem de cada linha do programa, incluindo todos os comentários dos vários níveis.
- c) Se o nível desejado for 1 ou 2, a listagem conterá as linhas de comentário extraídas do programa, conforme o nível desejado, pre-cedidas do número de linha correspondente à listagem do nível 3.

Por meio dos números de linhas pode-se estabelecer rapidamente uma correspondência visual entre as linhas das listagens dos vários níveis de documentação.

5. EXEMPLO. DETALHES DAS DOCUMENTAÇÕES.

No apêndice apresentamos como exemplo as listagens dos três níveis de documentação para o programa COMENT.PAS. Esse programa é justamente um listador para programas escritos em Pascal, COBOL ou ZIM. Ele foi escrito em Pascal para o VAX. A partir desse exemplo especificaremos aquilo que, até o presente momento, consideramos como interessante constar da documentação de cada nível.

5.1. Nível 1.

Note-se que ao programa principal e a cada procedimento corresponde um trecho da documentação deste nível. Esta deve ser totalmente conceitual, isto é, deve ser voltada para o problema (ou "mundo real", se for o caso). Assim, num sistema de conferência de notas fiscais colocaríamos "verificação do ICM" e não "verificação do campo NFICM do registro R2". Além disso, deve ser totalmente auto-contida, isto é, deve ser independente das documentações de níveis 2 e 3, a menos dos números de linhas. Denominamos nessa documentação de "entradas" e "saídas" os dados transportados de e para unidades externas, respectivamente. Denominamos de "importações" e "exportações" os dados que são globais a ou são parâmetros de procedimentos; no primeiro caso acrescentamos a palavra "globais".

Note-se, ainda, a enumeração dos nomes dos procedimentos chamados ("rotinas chamadas") e quais procedimentos chamam uma certa rotina ("chamado por").

5.2. Nível 2.

Observe-se que a documentação deste nível contém a de nível 1; distinguimos os comentários de níveis 1 e 2 por meio da impressão do dígito que constava do início do comentário correspondente.

Como no nível 1, os comentários de nível 2 devem ser totalmente conceituais. Além disso, devem ser totalmente independentes da documentação de nível 3, a menos dos números de linhas.

No exemplo, não colocamos as estruturas dos dados, pois elas são demasiado elementares; se existissem, deveriam ser independentes da estrutura "física". Por exemplo, deveriam comentar o mais conceitualmente possível o conteúdo dos arquivos como conjuntos ou tabelas (visão relacional) em relação com o problema ou "mundo real".

A estrutura dos programas é dada em um "Português estruturado". Chamamos a atenção para o fato de que procuramos evitar em nosso "Português estruturado" o uso de palavras reservadas das linguagens de programação usuais pois cremos que a

descrição de algoritmos deve libertar-se das amarras estruturais e linguísticas impostas por essas linguagens. Assim, não precedemos as condições com "se..." para evitar propositalmente a conotação com o 'if'. De fato, essas condições estão formuladas como perguntas com interrogação, e podem gerar tanto estruturas como 'if' quanto 'case'. Infelizmente, não encontramos uma alternativa melhor do que "caso contrário" para o "senão". É importante nesse nível tratar os dados como conjuntos e elementos de conjuntos. Por exemplo, uma certa malha de repetição ("loop") poderia ser especificada como "processa cada linha da nota fiscal".

Atenção especial deve ser dada para o alinhamento vertical: é ele que dá a estrutura de composição das ações. A regra empregada é a seguinte: se uma linha m começa na coluna i, a próxima linha n a ser executada sequencialmente depois de m, independentemente da composição desta, deve ser a próxima linha que também começa na coluna i; entre m e n não deve ocorrer nenhuma linha com algum caractere em colunas de índice menor ou igual a i.

Note-se a correspondência da numeração das documentações de nível 1 e de nível 2.

5.3. Nível 3.

Note-se que a documentação deste nível contém todo o programa, com os comentários de níveis 1, 2 e 3, exatamente como ocorrem no arquivo.

Observe-se que todas as variáveis do programa foram documentadas através de comentários de nível 3. Esse é o nível correto para isso, pois variáveis não têm nada a ver com a descrição conceitual dos níveis superiores. Além disso, foram colocados alguns comentários de nível 3 para comandos do programa.

Procuramos seguir a mesma regra de alinhamento vertical exposta para os comentários de nível 2. Infelizmente, o resultado não foi exatamente o desejado, mas é suficiente para ilustrar essa nossa única regra; a existência de uma só regra de alinhamento, ao contrário de maneiras usuais de alinhar comandos de programas, que contém regras específicas para certos comandos, tem a intenção de uniformizar e simplificar essa disposição, independentemente da linguagem.

Finalmente, note-se como os comandos do programa aparecem inseridos nos comentários de nível 2; cada um destes está documentando conceitualmente a ação executada pelos comandos que o seguem e que precedem o próximo comentário daquele nível.

6. DOCUMENTAÇÃO DAS ALTERAÇÕES.

Se se deseja documentar o histórico de todas as alterações efetuadas em um programa, a simples listagem das várias versões do mesmo não é prática, pois é relativamente difícil deduzir quais foram as modificações introduzidas empregando-se uma comparação visual entre as listagens de duas versões consecutivas. Muitos programadores empregam um programa especial de comparação de arquivos, que salienta as diferenças. Essa solução não é

satisfatória em casos de muitas alterações. Vejamos a solução que nosso sistema oferece para esse problema.

Afim de documentar as alterações projetamos um programa de listagem de alterações LA, que funciona da seguinte maneira:

- a) O usuário ativa LA e comunica a este que deseja iniciar alterações; fornece ainda o nome do arquivo, por exemplo A, onde está seu programa P e o nome do editor de textos E que deseja utilizar para editar as alterações.
- b) LA abre um novo arquivo A1 e copia P em A1, numerando suas linhas..
- c) LA passa o controle ao editor E dando A1 como origem do texto.
- d) O usuário altera A1 empregando E, sem alterar nenhuma numeração de linhas; ele pode alterar o conteúdo de linhas de P, inserir neste novas linhas (sem colocar a numeração destas) ou eliminar linhas (eliminando também sua numeração). O usuário deve também inserir linhas de comentário especiais no início de A1, usando um código de comentário especial como '/' a partir da coluna 1; essas linhas servem para documentar-se a razão de se ter feito as alterações. O programa alterado, que denominaremos de P1, é então gravado em A1.
- e) O usuário ativa LA novamente e comunica a este que deseja emitir as listagens de alterações, fornecendo os nomes de A e de A1.
- f) LA varre A numerando internamente as linhas de P. Em paralelo, lê A1 comparando as suas linhas com as de mesmo número de A. Se houver alguma diferença de conteúdo entre duas linhas de mesmo número, ambas são gravadas em um arquivo de saída S, a de A precedida da mensagem "original" e a de A1 de "alterada". Se houver alguma linha de A com um número de linha que não ocorre em A1, ela é gravada em S precedida de "eliminada". Se ocorrer uma sequência de linhas de A1 sem numeração, estas são gravadas em S precedidas da mensagem "linhas inseridas", da linha de A correspondente à linha numerada de A1 imediatamente anterior à sequência, e seguidas pela linha de A imediatamente seguinte. Todas as linhas de A1 com código de comentário especial ('/') são também inseridas em S.
- g) À medida que LA varre A1 no passo anterior, vai regravando A1 retirando as numerações de linhas e eliminando as linhas com código de comentário especial ('/').
- h) Ao terminar a varredura de A e de A1, LA executa uma rotina de listagem semelhante ao listador, que lê S e produz a impressão das alterações em todos os níveis de documentação; cada nível é precedido dos comentários especiais ('/').

Como resultado desses procedimentos permanece o programa P em A, P1 é criado em A1 e obtém-se a listagem das alterações efetuadas na passagem de P para P1, divididas nos vários níveis de documentação, com as numerações das linhas de P para referência com o programa original e precedida dos comentários especiais que descrevem os motivos das alterações. O arquivo S pode ser novamente

empregado para obter-se novas cópias das alterações, bastando para isso ativar LA e informar que se deseja executar esse passo.

Note-se que esse sistema de documentar as alterações serve tanto para registrar-se as alterações de manutenção, como também para documentar os passos de desenvolvimento de um programa em um método por refinamentos sucessivos.

Existem vários sistemas de documentação de alterações, denominados de "version control systems"; uma descrição de vários desses sistemas pode ser encontrada em [1]. Em particular, o sistema DIFF originário do UNIX e transposto para o TURBO-C é muito empregado no IBM-PC. Cremos que nossa proposta é superior a esses produtos pois está voltada especificamente para o sistema de documentação aqui descrito, além de adaptar-se a qualquer editor de textos. Note-se que a numeração das linhas feitas pelo nosso sistema resolve todos os problemas de comparação entre as linhas sem prejudicar o usuário, pelo contrário, está dentro do esquema da documentação produzida pelo listador.

6. RESULTADOS.

Com o método aqui exposto pode-se obter os seguintes resultados:

- R1. Documentação de um programa, do seu desenvolvimento e de sua manutenção em vários níveis de abstração.
- R2. Programação "top-down". Deve-se desenvolver um programa começando-se pelos comentários de nível 1, a seguir inserir os de nível 2, e finalmente o programa propriamente dito e os comentários do nível 3.
- R3. Geração automática de documentação.
- R4. Aceleração do desenvolvimento. A documentação em vários níveis auxilia a programação e diminui o tempo dedicado aos testes.
- R5. Aceleração da manutenção. Para se fazer manutenção de um programa, pode-se localizar o ponto de alteração (ou de erro) "navegando-se" pelos níveis decrescentes de abstração da documentação. Os números de linhas podem ser empregados para uma rápida movimentação de um nível para outro.
- R6. Atualização permanente da documentação. Qualquer alteração do programa deve redundar em alteração nos comentários de níveis 3, 2 e mesmo 1 se necessário. Os listadores de alterações e de documentação produzirão a documentação das alterações efetuadas e da nova versão da documentação do programa.
- R7. Programação encadeada. Várias pessoas podem participar do desenvolvimento de um programa, particionando-se a equipe conforme os três níveis de abstração. Isso garante a qualidade das documentações de níveis 1 e 2.
- R8. Controle da programação "top down". A listagem das alterações correspondendo ao desenvolvimento "top down", contendo as várias versões do refinamento sucessivo, possibilita à gerência verificar se a metodologia está sendo corretamente seguida.
- R9. Controle da documentação. A gerência pode controlar a documentação dos programas, já que é relativamente fácil seguir

as documentações de níveis 1 e 2. A partir destas, não deve ser difícil seguir o programa distribuído entre elas, e seus comentários de detalhe. A documentação das alterações permite controlar a manutenção das documentações dos níveis mais altos.

R10. Aumento da portabilidade. Alterações de programas devidas a mudanças de ambiente de processamento podem ser feitas com muito maior rapidez, pois devem afetar muito pouco os níveis 1 e 2 de documentação. Em particular, comentários de nível 3 podem ser inseridos para salientar as dependências de trechos do programa em relação ao ambiente.

R11. Documentação independente dos detalhes de programação. A documentação de nível 1 pode ser empregada por pessoas que desejam conhecer a funcionalidade dos programas e das rotinas sem se importar com detalhes de programação.

7. CONCLUSÕES.

Apresentamos aqui um método extremamente simples de produzir e manter a documentação de programas. Estamos cientes de que esta não é uma área promissora, pois a programação está desaparecendo devido à introdução de geradores universais de aplicações para processamento de dados administrativos, que em alguns casos como o sistema LDT geram automaticamente programas e documentação, sendo esta de altíssimo nível [2]. No entanto, sobram ainda todos os desenvolvimentos de "software" básico, quando não são usadas técnicas formais como VDM (ver, como exemplo dessa técnica, [3]), e a grande maioria dos sistemas desenvolvidos em linguagens de programação algorítmicas e de "4a. geração". Outros esforços têm sido produzidos no sentido de se introduzir comentários significativos nos programas, como o sistema WEB de Knuth [4]. No entanto, trata-se de sistema muito mais complexo, abrangendo inclusive a parte de asserções para prova de corretude (para uma extensa referência a respeito, ver [5]). Essas asserções podem evidentemente ser introduzidas em nosso método como comentários de nível 2. O mesmo se passa com comentários de tipo formal como os propostos por Krieg-Brueckner e Luckham [6]. Nesse dois casos, cremos que as asserções ficam ligadas demais ao código, de modo que poderiam constituir um nível adicional, entre os nossos níveis 2 e 3. Com isso as asserções teriam seus próprios comentários; estes serviriam para esclarecer em um nível informal o conteúdo matemático daquelas, o que daria a documentação do desenvolvimento delas e do programa. A documentação desse desenvolvimento não é a intenção exposta em [6]. Além disso, nosso sistema tem a vantagem adicional de documentar as versões.

É interessante observar que as considerações de Beckman [7] contrárias ao uso de comentários em programas não se aplicam às propostas aqui apresentadas, pois os comentários que propomos estão intimamente ligados à funcionalidade dos programas e de seus trechos, ao desenvolvimento dos mesmos e ao uso do nível 1 independentemente dos outros níveis.

Testamos e desenvolvemos estas idéias com alunos de nossa disciplina de Compilação, em que eles elaboram individualmente um

programa de milhares de linhas. Apesar do protesto inicial dos alunos, estes logo perceberam as enormes vantagens do sistema de documentação. Em 1987, apenas 22% da turma concluiu o projeto durante o semestre; em 1988, na turma em que testamos o método, 65% dos alunos o concluiu, provavelmente devido, em boa parte, à documentação.

Várias melhorias podem ser introduzidas no sistema. Por exemplo, pode-se eliminar o aparecimento dos números dos níveis 1 e 2 nas listagens desses níveis, bastando para isso enquadrar totalmente os comentários de nível 1 do programa principal e de cada procedimento em um retângulo de asteriscos. Como vimos, a possibilidade de se documentar as alterações de uma versão para outra pode servir de base para documentar todo processo de desenvolvimento do programa por refinamentos sucessivos; além disso os comentários de nível 2 poderiam ser subdivididos em vários subníveis, para produzir uma documentação completa desse desenvolvimento.

O colega Arnaldo Mandel propôs um sistema de 5 níveis. O nível 1 contém a documentação para o usuário "externo", do sistema como um todo, como se ele só tivesse o programa-objeto; o nível 2 contém informações para a compilação; o nível 3 é uma documentação para reutilização das rotinas, com variáveis globais, parâmetros, modo de chamar os procedimentos, etc.; o nível 4 é correspondente ao nosso nível 2; o nível 5 contém o histórico de alterações correspondente aos nossos comentários especiais (ele não propõe que se documente as alterações propriamente ditas).

Roberto C. Mayer implementou o listador em C, possibilitando que a especificação de nível (1 ou 2) valha para uma sequência de linhas, e não somente para cada linha: o encerramento do comentário fecha a sequência. Além disso, sua codificação de comentário de um desses níveis pode ser colocada em qualquer posição de uma linha, ao contrário da nossa proposta, que exige essa colocação no início da linha, simplificando a construção do listador. Seu sistema exige, obviamente, um analisador léxico para detectar os níveis e o encerramento dos comentários correspondentes.

Mirza Neuman propôs o armazenamento apenas da última versão da documentação e dos arquivos com as alterações de uma versão para outra, gerando-se uma versão anterior a partir desses dados. Parece-nos que é possível construir-se esse programa gerador, pois todos os dados necessários estão disponíveis.

8. AGRADECIMENTOS.

O sistema aqui apresentado foi desenvolvido a partir de idéias lançadas por Marcos Ximenes para o sistema ZIM. Especial agradecimento é devido a Afonso C.R. Mastrelli, que programou o exemplo dado no apêndice, com enorme paciência de introduzir paulatinamente as nossas seguidas sugestões. Wagner T. Martins programou uma excelente versão do listador em NATURAL para o sistema ADABAS.

9. REFERÊNCIAS.

- [1] Rex, J. - Keeping Track: Five Version Control Systems - Computer Language 5, 6 (June 1988), pp. 111-122.
- [2] Setzer, V.W. e Marussi, E. - LDT: Um Gerador Universal de Aplicações para Processamento de Dados - RT-MAC-8806, Departamento de Ciência da Computação do IME-USP (junho de 1988).
- [3] Bjorner, D. - Programming Languages: Formal Development of Interpreters and Compilers - in Proc. Intl. Comp. Symp. ICS'77, North Holland Publ., pp. 1-21.
- [4] Knuth, D. - Literate Programming - Computer Journal 27, 2 (1984), pp. 97-111.
- [5] Gries, D. - The Science of Programming - Springer Verlag, N.York (1981).
- [6] Krieg-Brueckner, B. e Luckham, D.C. - Anna: Towards a Language for Annotating Ada Programs - ACM SIGPLAN Notices 15, 11 (Nov. 1980), pp. 128-138.
- [7] Beckman, A. - Comments Considered Harmful - ACM SIGPLAN Notices 12, 4 (April 1977), pp. 94-97.

PROGRAMA: COMENT.PAS DOC.NIVEL 1 DATA: 8-AUG-1988 15:57 PAG: 1

```

1 *****
2 1 TITULO: COMENTARIO
3 1 AUTOR : AFONSO CELSO ROCHA MASTRELLI
4 1 DATA : 29/06/88
5 1 VERSAO: 1.2
6 1 FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-
7 1 GRAMA:
8 1 ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)
9 1 COMENTARIO POR NIVEIS: CODIGO DO NIVEL DE COMENTARIO DESEJADO:
10 1 SAIDAS : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE
11 1 ROTINAS CHAMADAS: LIGOU-COMMAND (TEXTUAL), MOSTRA-TELA, CABECALHO,
12 1 RELATOS, RELATPAS E SYSSASCTM (TEXTUAL)
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
36 1
37 1
38 1
39 1
40 1
41 1
42 1
43 1
44 1
45 1
46 1
47 1
48 1
49 1
50 1
51 1
52 1
53 1
54 1
55 1
56 1
57 1
58 1
59 1
60 1
61 1
62 1
63 1
64 1
65 1
66 1
67 1
68 1
69 1
70 1
71 1
72 1
73 1
74 1
75 1
76 1
77 1
78 1
79 1
80 1
81 1
82 1
83 1
84 1
85 1
86 1
87 1
88 1
89 1
90 1
91 1
92 1
93 1
94 1
95 1
96 1
97 1
98 1
99 1
100 1
101 1
102 1
103 1
104 1
105 1
106 1
107 1
108 1
109 1
110 1
111 1
112 1
113 1
114 1
115 1
116 1
117 1
118 1
119 1
120 1
121 1
122 1
123 1
124 1
125 1
126 1
127 1
128 1
129 1
130 1
131 1
132 1
133 1
134 1
135 1
136 1
137 1
138 1
139 1
140 1
141 1
142 1
143 1
144 1
145 1
146 1
147 1
148 1
149 1
150 1
151 1
152 1
153 1
154 1
155 1
156 1
157 1
158 1
159 1
160 1
161 1
162 1
163 1
164 1
165 1
166 1
167 1
168 1
169 1
170 1
171 1
172 1
173 1
174 1
175 1
176 1
177 1
178 1
179 1
180 1
181 1
182 1
183 1
184 1
185 1
186 1
187 1
188 1
189 1
190 1
191 1
192 1
193 1
194 1
195 1
196 1
197 1
198 1
199 1
200 1
201 1
202 1
203 1
204 1
205 1
206 1
207 1
208 1
209 1
210 1
211 1
212 1

```

```

1 1=====
2 1 TITULO: COMENTARIO
3 1 AUTOR : AFONSO CELSO ROCHA MASTRELLI
4 1 DATA : 29/06/88
5 1 VERSAO: 1.2
6 1 FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-
7 1 GRAMA
8 1 ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)
9 1 COMENTARIO POR NIVEIS: CODIGO DO NIVEL DE COMENTARIO DESEJADO;
10 1 SAIDAS : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE;
11 1 ROTINAS CHAMADAS: LISDSO,COMAND (EXTERNAL), MOSTRA-TELA, CABECALHO,
12 1 RELA , RELATPS E SYSSASCTIM (EXTERNAL);
13 1=====
51 1 ROTINA: MOSTRA-TELA
52 1 FINALIDADE: MOSTRA MENU AO USUARIO, PERGUNTANDO E RECEBENDO O NIVEL DE
53 1 DOCUMENTACAO QUE ELE QUER E O NOME DO ARQUIVO A SER LIDO,
54 1 PASSANDO ESSAS INFORMACOES AO PROGRAMA PRINCIPAL;
55 1 ENTRADAS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
56 1 EXPORTACOES GLOBAIS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO;
57 1 CHAMADO POR: COMENTARIO;
58 1=====
59 1 INICIO DE MOSTRA-TELA
61 2 LIMPA A TELA
63 2 MOSTRA MENU DE OPCOES AO USUARIO
65 2 LE OPCAO DO USUARIO
84 2 OPCAO LIDA E' DE CONTINUAR A EXECUCAO ?
86 2 LE NOME DO ARQUIVO DE ENTRADA
91 2 LE NOME DA LINGUAGEM UTILIZADA NO PROGRAMA
95 2 LINGUAGEM NAO E' COBOL NEM ZIM?
97 2 ASSUME-SE PASCAL COMO LINGUAGEM UTILIZADA
101 2 FIM-DE-MOSTRA-TELA
106 2=====
111 1 ROTINA: CALECALHO;
112 1 FINALIDADE: IMPRIMIR CABECALHO NO INICIO DE CADA PAGINA DO RELATORIO;
113 1 IMPORTACOES GLOBAIS: NOME DO ARQUIVO LIDO, NIVEL DE DOCUMENTACAO,
114 1 DATA E HORA ATUAIS;
115 1 SAIDA: LINHA DE CABECALHO IMPRESSA NO ALTO DE CADA PAG. DA LISTAGEM;
116 1 CHAMADO POR: COMENTARIO;
117 1=====
118 1 INICIO DE LABCALHO;
121 2 ATUALIZA O NUMERO DA PAGINA.
122 2 PAGINA ESTA' CHEIA ?
125 2 RELATORIO JA' FOI INICIALIZADO ?
127 2 SALTA PAGINA
129 2 ESCREVE LINHA DE CABECALHO
131 2 FIM DE CABECALHO;
138 2=====
142 1 ROTINA: RELAT;
143 1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM COBOL OU ZIM) E MONTA O
144 1 DE SAIDA;
145 1 IMPORTACOES: CARACTERE QUE IDENTIFICA A LINHA DE COMENTARIO NA LINGUAGEM.
146 1 UTILIZADA (COBOL OU ZIM);
147 1=====

```



```

PROGRAMA: COMENT-PAS          DOC=NIVEL 3    DATA: 8-FUG-1988 10:00    PAG: 1
1 (C)=====
2 (C) TITULO: COMENTARIO
3 (C) AUTOR : AFONSO CELSO ROCHA MASTRELLI
4 (C) DATA : 24/06/88
5 (C) VERSAO: 1.2
6 (C) FINALIDADE : GERAR RELATORIO CONTENDO A DOCUMENTACAO DE DETERMINADO PRO-
7 (C) GRAMA
8 (C) ENTRADAS : ARQUIVO QUE CONTEM O CODIGO FONTE (EM PASCAL, COBOL OU ZIM)
9 (C) COMENTADO POR NIVEIS: CODIGO DO NIVEL DE COMENTARIO DESEJADO:
10 (C) SAIDAS : RELATORIO COM A DOCUMENTACAO EXTRAIDA DO ARQUIVO FONTE;
11 (C) ROTINAS CHAMADAS: LIB$DO-COMMAND (EXTERNAL), MOSTRA-TELA, CABECALHO,
12 (C) RELAT, RELATPAS E SYSSASCTIM (EXTERNAL);
13 (C)=====
14 PROGRAM COMENTARIO(INPUT,OUTPUT);
15
16 CONST
17 CLS = CHR(27) + 'C'; CHR(27) + 'CJ'; (C) LINHA A TELA DO VT100/200/220 *)
18 TYPE
19 LINHA = VARYING(120) OF CHAR; (C) PARA LINHA DE IMPRESSAO;
20 CARACTERE = CHAR; (C) PARA PARAMETRO DE RELAT;
21 $WORD = $WORD(0..45535); (C) PARA PARAMETRO DE SYSSASCTIM;
22 $DATE_TIME = $DATE_TIME(0..45535); (C) PARA PARAMETRO DE SYSSASCTIM;
23
24 VAR
25 COMANDO : VARYING(200) OF CHAR; (C) DURANTE OCL A SER EXECUTADO;
26 LIMPROG : LINHA; (C) GUARDA LINHA DO PROGRAMA LIDO;
27 OPCAO : INTEGER; (C) MRO DA OPCAO DO USUARIO NO MENU;
28 MONERO : VARYING(30) OF CHAR; (C) NOME DO ARQUIVO CUC CONTEM O PROGRAMA;
29 SAIDA : TEXT; (C) CONTEM O RELATORIO DE SAIDA;
30 ENTRADA : TEXT; (C) PROGRAMA COMENTADO A SER LIDO;
31 LING : CARACTERE; (C) LINGUAGEM DO PROGRAMA A SER LIDO;
32 NUNLIN : INTEGER; (C) NUMERACAO DAS LINHAS DO FONTE LIDO;
33 LIMPAC : INTEGER; (C) NUMERO DE LINHAS POR PAGINA;
34 MPAG : INTEGER; (C) NUMERO DA PAGINA DO RELATORIO;
35 $DATA_HORA : VARYING(17) OF CHAR; (C) DATA E HORA RECEBIDA DO SISTEMA;
36 RETORNO : INTEGER; (C) SOMENTE ACHILIA O RECEBIMENTO DA HORA;
37
38
39 (C) ROTINA EXTERNA QUE EXECUTA COMANDO OCL
40 PROCEDURE LIB$DO-COMMAND(CMDTAT : VARYING(43) OF CHAR); EXTERN;
41
42
43 (C) FUNCAO EXTERNA QUE RETORNA DATA E HORA ATUAIS DO SISTEMA OPERACIONAL
44 EXTERNAL FUNCTION SYSSASCTIM(REF NADA1 : $WORD := $IMMED 0;
45 DATA_HORA : $CLASS-S2
46 XREF NADA2 : $DATE_TIME := $IMMED 0;
47 NADA3 : UNSIGNED := $IMMED 0); INTEGER; EXTERN;
48
49
50
51 (C)=====
52 (C) ROTINA: MOSTRA-TELA;
53 (C) FINALIDADE: MOSTRA MENU AO USUARIO, PERGUNTANDO E RECEBENDO O NIVEL DE

```

```

PROGRAMA3 COMENT.PAS      DOC.NIVEL 3      DATA: 8-AUG-1988 14:00      PAG: 2

54 (*1 DOCUMENTACAO QUE ELE QUER E O NOME DO ARQUIVO A SER LIDO, *)
55 (*1 PASSANDO ESSAS INFORMACOES AO PROGRAMA PRINCIPAL *)
56 (*1 ENTRADAS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO *)
57 (*1 EXPORTACOES GLOBAIS: NIVEL DE DOCUMENTACAO E NOME DO ARQUIVO A SER LIDO *)
58 (*1 CHAMADO POR: COMENTARIOS *)
59 (*1 ***** *)
60 PROCEDURE MOSTRA_TELA;
61 (*2 INICIO DE MOSTRA_TELA *)
62 BEGIN
63 (*2 LIMPA A TELA *)
64   WRITE(CLS);
65 (*2 MOSTRA MENU DE OPCOES AO USUARIO *)
66   WRITE(' :25, "DOCUMENTACAO DE SOFTWARE" ');
67   WRITE(' :25, ***** ');
68   WRITE;
69   WRITE(' :10, "VOCE PODE LISTAR 3 NIVEIS DE DOCUMENTACAO DE UM PROGRAMA" ');
70   WRITE(' :10, "O PROGRAMA PODE ESTAR NUMA DAS SEGUINTE LINGUAGENS: " ');
71   WRITE;
72   WRITE(' :27, "PASCAL (DEFAULT)" ');
73   WRITE(' :27, "COBOL" ');
74   WRITE(' :27, "ZIM" ');
75   WRITE;
76   WRITE(' :26, "NIVEIS DE DOCUMENTACAO" ');
77   WRITE;
78   WRITE(' :12, "1. DESCRICAO FUNCIONAL + FINALIDADES" ');
79   WRITE(' :12, "2. NIVEL 1 + PSEUDO-CODIGO" ');
80   WRITE(' :12, "3. NIVEIS 1 E 2 + CODIGO FONTE" ');
81   WRITE(' :12, "4. F I M" ');
82   WRITE;
83   WRITE(' :15, "SUA OPCAO E" : ');
84 (*2 LE OPCAO DO USUARIO *)
85   READLN(OPCAO);
86 (*2 OPCAO LIDA E SE CONTINUAR A EXECUCAO ? *)
87   IF (OPCAO = 1) OR (OPCAO = 2) OR (OPCAO = 3) THEN
88     BEGIN
89       WRITE;
90       WRITE(' :15, "NOME DO ARQUIVO : ');
91       LE NOME DO ARQUIVO DE ENTRADA
92       READLN(NOMEARQ);
93       WRITE;
94       WRITE(' :15, "LINGUAGEM UTILIZADA : ');
95       LE NOME DA LINGUAGEM UTILIZADA NO PROGRAMA
96       READLN(LING);
97       LINGUAGEM NAO E COBOL NEM ZIM?
98       CASE LING OF
99         'C', 'C' : LING := 'C';
100        '2', '2' : LING := '2';
101        ASSUME-SE PASCAL COMO LINGUAGEM UTILIZADA
102        OTHERWISE
103          LING := 'P';
104        END; (* CASE *)
105      END; (* IF *)
106 (*2 FIM-DE-MOSTRA-TELA *)
107 END; (* MOSTRA TELA *)

```

[illegible]

```

PROGRAMA: COMENT.PAS          DOC=NIVEL 3    DATA: 8-AUG-1988 16:00    PAG: 4

162 (*2 LE LINHA SEQUENCIALMENTE
163 READLN(ENTRADA,LINPROG);
164 (*2 NUMERA A LINHA LIDA
165 NUMLIN := NUMLIN + 1;
166 (*2 DOCUMENTACAO E' DE NIVEL 3 ?
167 IF OPCAO = 3 THEN
168 BEGIN
169 ESCREVE LINHA NO RELATORIO
170 LINPAG := LINPAG + 1;
171 CABECALHO;
172 WRITE(LINSAIDA, ' ', NUMLIN:5, ' ', LINPROG);
173 END
174 (*2 CASO CONTRARIO
175 ELSE
176 BEGIN
177 LINHA NAO E' NULA ?
178 IF (LENGTH(LINPROG) > 2) THEN
179 BEGIN
180 LINHA E' DE COMENTARIO NIVEL 1 ?
181 IF (SUBSTR(LINPROG,1,2) = IDENT1) THEN
182 BEGIN
183 LINPAG := LINPAG + 1;
184 CABECALHO;
185 (*2 ESCREVE LINHA NO RELATORIO
186 WRITE(LINSAIDA, ' ', NUMLIN:5, ' ', SUBSTR(LINPROG,2,LENGTH(LINPROG)-1));
187 END
188 (*2 CASO CONTRARIO
189 ELSE
190 (*2 E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?
191 IF ((SUBSTR(LINPROG,1,2) = IDENT2) AND (OPCAO = 2)) THEN
192 BEGIN
193 LINPAG := LINPAG + 1;
194 CABECALHO;
195 (*2 ESCREVE LINHA NO RELATORIO
196 WRITE(LINSAIDA, ' ', NUMLIN:5, ' ', SUBSTR(LINPROG,2,LENGTH(LINPROG)-1));
197 END;
198 END; (* IF *)
199 END; (* IF *)
200 END; (* WHILE *)
201 WRITE(LINSAIDA);
202 (*2 FIM DE RELAT
203 END; (* RELAT *)
204
205 (*****
206 (*1 ROTINA: RELATPAS;
207 (*1 FINALIDADE: FAZ LEITURA DO ARQUIVO DE ENTRADA (EM PASCAL) E MONTA O DE
208 (*1 SAIDA:
209 (*1 SAIDA: COMENTARIOS DO PROGRAMA DE ENTRADA NO NIVEL DESEJADO;
210 (*1 CHAMADO POR: COMENTARIO1
211 (*1 INICIO DE RELATPAS
212 PROCEDURE RELATPAS;
213 BEGIN
214 (*2 INICIO DE RELATPAS
215 BEGIN

```

```

PROGRAMA: COMENT.PAS          DOC-NIVEL 3      DATA: 8-AUG-1988 16:00      PAG: 5
216 (*2 PARA CADA LINHA DO ARQUIVO DE ENTRADA :
217 WHILE NOT EOF(ENTRADA) DO
218 BEGIN
219 (*2
220 LE LINHA SEQUENCIALMENTE
221 READLN(ENTRADA,LINPROG);
222 (*2
223 MUMLIN := MUMLIN + 1;
224 DOCUMENTACAO E: DE NIVEL 3 ?
225 IF OPCAD = 3 THEN
226 BEGIN
227 ESCREVE LINHA NO RELATORIO
228 LINPAG := LINPAG + 1;
229 CABECALHO;
230 WRITELN(SAIDA, ' :',MUMLIN:5, ' ',LINPROG);
231 END
232 (*2
233 CASO CONTRARIO
234 ELSE
235 BEGIN
236 LINHA NAO E' NULA ?
237 IF (LENGTH(LINPROG) > 3) THEN
238 BEGIN
239 LINHA E' DE COMENTARIO NIVEL 1 ?
240 IF (SUBSTR(LINPROG,1,3) = '(,1)') THEN
241 BEGIN
242 LINPAG := LINPAG + 1;
243 CABECALHO;
244 ESCREVE LINHA NO RELATORIO
245 WRITELN(SAIDA, ' :',MUMLIN:5, ' ',SUBSTR(LINPROG,3,INDEX(LINPROG,':')-3));
246 END
247 (*2
248 CASO CONTRARIO
249 ELSE
250 BEGIN
251 E' DE COMENTARIO NIVEL 2 E USUARIO QUER NIVEL 2 ?
252 IF ((SUBSTR(LINPROG,1,3) = '(,2)') AND (OPCAD = 2)) THEN
253 BEGIN
254 LINPAG := LINPAG + 1;
255 CABECALHO;
256 ESCREVE LINHA NO RELATORIO
257 WRITELN(SAIDA, ' :',MUMLIN:5, ' ',SUBSTR(LINPROG,3,INDEX(LINPROG,':')-3));
258 END;
259 (*2
260 ENDS (* IF *)
261 ENDS (* WHILE *)
262 WRITELN(SAIDA);
263 FIM DE RELATPAS
264 END; (* RELATPAS *)
265
266 (*2
267 PROGRAM PRINCIPAL
268 BEGIN
269 OPCAD := 01

```

PROGRAMA: COMENT.PAS

DOC.NIVEL 3 DATA: 8-AUG-1988 16:00

PAG: 6

```

270 DATA-MORA := '
271 MUNIM := 0;
272 EXIBE TELA DE ENTRADA
273 MOSTRA-TELA;
274 (* OPCAO DO USUARIO E* CONTINUAR ?
275 IF (OPCAO > 0) AND (OPCAO < 4)) THEN
276 BEGIN
277 ABRE ARQUIVOS DE ENTRADA E SAIDA
278 OPENTENTRADA;MONTEAR;
279 HISTORY := OLD;
280 ACCESS-METNOD := SEQUENTIAL;
281 OPEN(SAIDA,'COMMENT.LIS',MISTORY := NEW);
282 RESET(CENTRADA);
283 REMRITE(SAIDA);
284 RETORNO := SYSSASCTM(DATA-MORA,);
285 (* FORÇA A IMPRESSAO DO CABECALHO NA PRIMEIRA PAGINA *)
286 LIMPAG := 70;
287 CABECALHO;
288 ENO; (* IF *)
289 (* OPCAO DO USUARIO E* NIVEL 1 OU 2 ?
290 CASE OPCAO OF
291 1,2,3 : BEGIN
292 IF (MON .RO-LENGTH > 20) THEN
293 MONTEAR := SUBSTR(MONTEAR,1,20);
294 ENTRADA ESTA* EM PASCAL ?
295 IF (LING = 'P') THEN
296 PROCSSA ARQUIVO EM PASCAL
297 RELATPAS
298 CASO CONTRARIO
299 ELSE
300 PROCSSA ARQUIVO EM COBOL OU ZIM
301 IF (LING = 'C') THEN
302 RELAT('C')
303 ELSE
304 RELAT('Z');
305 WANDA RELATORIO PARA A FILA DE IMPRESSAO
306 COMANDO := 'PRINT COMMENT.LIS'; *)
307 LIBSOO-COMMAND(COMANDO);
308 ENO; (* 1,2,3 *)
309 ENO; (* CASE *)
310 ALGUM ARQUIVO FOI ABERTO ?
311 IF (OPCAO > 0) AND (OPCAO < 4)) THEN
312 BEGIN
313 FECHA ARQUIVOS UTILIZADOS
314 CLOSE(CENTRADA);
315 CLOSE(SAIDA);
316 ENO;
317 (* FIM DE COMENTARIO
318 END. (* COMENTARIO *)

```

**RELATÓRIOS TÉCNICOS
TÍTULOS PUBLICADOS**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA USP**

RT-MAP-7702 - (Dezembro 1977)

V.W. Setzer

A Note on a Recursive Top-Down Analyzer of N.Wirt

RT-MAP-7704 - (Dezembro 1977)

V.W. Setzer, M.M. Sanches

A linguagem "LEAL" para Ensino básico de Computação

RT-MAP-7802 - (Fevereiro 1978)

Silvio Ursic, Cyro Patarra

**Exact solution of Systems of Linear Equations with
Interactive Methods**

RT-MAC-7803 - (Março 1978)

Martin Groetschel, Yoshiko Wakabayashi

Hypohamiltonian Digraphs

RT-MAP-7804 - (Maio 1978)

Martin Groetschel, Yoshiko Wakabayashi

Hypotractable Digraphs

RT-MAP-7805 - (Junho 1978)

U. Hesse, V.W. Setzer

**The Line-Justifier: an example of program development by
transformations**

RT-MAP-7809 - (Novembro 1978)

V.W. Setzer

**Program development by transformations applied to relational
Data-Base queries**

RT-MAP-7811 - (Novembro 1978)

D.T. Fernandes e C. Patarra

Sistemas Lineares Esparsos, um Método Exato de Solução

RT-MAP-7812 - (Novembro 1978)

V.W. Setzer e G. Bressan

Desenvolvimento de Programas por Transformações: uma
Comparação entre dois Métodos

RT-MAP-7814 - (Dezembro 1978)

Martin Groetschel e Yoshiko Wakabayashi

On the Complexity of the Monotone Asymmetric Travelling
Salesman Polytope I: HYPOHAMILTONIAN FACETS

RT-MAP-7901 - (Fevereiro 1979)

Martin Groetschel, Yoshiko Wakabayashi

On the Complexity of the Monotone Asymmetric Travelling
Salesman Polytope II: HYPOTRACEABLE FACETS

RT-MAP-7902 - (Junho 1979)

M.M. Sanches e V.W. Setzer

A Portabilidade do Compilador para a Linguagem LEAL

RT-MAP-7903 - (Julho 1979)

Martin Groetschel, Carsten Thomassen, Yoshiko Wakabayashi
Hypotractable Digraphs

RT-MAP-8003 - (Setembro 1980)

Routo Terada

Fast Algorithms for NP-Hard Problems which are Optimal or
near-Optimal with Probability one

RT-MAP-8004 - (Setembro 1980)

V.W. Setzer e R. Lapyda

Uma Metodologia de Projeto de Bancos de Dados para o Sistema
ADABAS

RT-MAP-8005 - (Outubro 1980)

Imre Simon

On Brzozowski's Problem: $(1 \cup A)^* = A^*$

RT-MAP-8101 - (Fevereiro 1981)

Luzia Kazuko Yoshida e Gabriel Richard Bitran

Um Algoritmo para Problemas de Programação com Variáveis
Zero-Um

RT-MAP-8103 - (Abril 1981)

V.W. Setzer, R. Lapyda

Design of Data Models for the ADABAS System using the
Entity-Relationship Approach

RT-MAP-8105 - (Maio 1981)

U.S.R. Murty

Projective Geometries and their Truncations

RT-MAP-8106 - (Junho 1981)

V.W. Setzer, R. Lapyda

Projeto de Bancos de Dados, Usando Modelos Conceituais
(Este Relatório Técnico complementa o RT-MAP-8103. Ambos
substituem o RT-MAP-8004 ampliando os conceitos ali
expostos.)

RT-MAP-8107 - (Agosto 1981)

Maria Angela Gurgel, Yoshiko Wakabayashi

Embedding of Trees

RT-MAP-8201 - (Janeiro 1982)

Siang Wun Song

On a High-Performance VLSI Solution to Database Problems

RT-MAP-8202 - (Junho 1982)

Maria Angela Gurgel, Yoshiko Wakabayashi

A Result on Hamilton-Connected Graphs

RT-MAP-8205 - (Junho 1982)

Arnaldo Mandel

Topology of Oriented Matroids

RT-MAP-8206 - (Novembro 1982)

Erich J. Neuhold

Database Management Systems: A General Introduction

RT-MAP-8207 - (Novembro 1982)

Béla Bollobás

The Evolution of Random Graphs

RT-MAP-8208 - (Novembro 1982)

V.W. Setzer

Um Grafo Sintático para a Linguagem PL/M-80

RT-MAP-8209 - (Novembro 1982)

Jayme Luiz Szwarcfiter

A Sufficient Condition for Hamilton Cycles

RT-MAP-8302 - (Fevereiro 1983)

Béla Bollobás, Istvan Simon

Repeated Random Insertion into a Priority Queue

RT-MAP-8303 - (Julho 1983)

V.W. Setzer, P.C.D. Freitas e B.C.A. Cunha

Um Banco de Dados de Medicamentos

RT-MAP-8305 - (Julho 1983)

Arnaldo Mandel

The 1-Skeleton of Polytopes, oriented Matroids and some
other lattices

RT-MAP-8306 - (Agosto 1983)

Arnaldo Mandel

Alguns Problemas de Enumeração em Geometria

RT-MAP-8307 - (Agosto 1983)

Siang Wun Song

Complexidade de E/S e Projetos Optimais de Dispositivos para
Ordenação

RT-MAP-8402 - (Abril 1984)

V.W. Setzer

Manifesto contra o uso de computadores no Ensino de 12 Grau

RT-MAP-8404 - (Setembro 1984) (7 pgs)

Imre Simon

A Factorization of Infinite Words

RT-MAP-8405 - (Setembro 1984) (6 pgs)

Imre Simon

The Subword Structure of a Free Monoid

RT-MAP-8406 - (Setembro 1984) (25 pgs)

Jairo Z. Gonçalves e Arnaldo Mandel

Are There Free Groups in Division Rings?

RT-MAP-8407 - (Novembro 1984) (18 pgs)

Paulo Feofiloff and D.H. Younger

Vertex-Constrained Transversals in a Bipartite Graph

RT-MAP-8408 - (Novembro 1984) (126 pgs)

Paulo Feofiloff

Disjoint Transversals of Directed Coboundaries

RT-MAP-8409 - (Novembro 1984) (16 pgs)

Paulo Feofiloff e D.H. Younger

Directed cut transversal packing for source-sink connected
graphs

RT-MAP-8501 - (Maio 1985) (11 pgs)

Siang Wun Song

Disposições Compactas de Árvores no Plano

RT-MAP-8502 - (Julho 1985) (11 pgs)

Paulo Feofiloff

Transversais de Cortes Orientados em Grafos Bipartidos

RT-MAP-8504 - (Setembro 1985) (110 pgs)

Christian Choffrut

Free Partially Commutative Monoids

- RT-MAP-8505 - (Outubro 1985) (40 pgs)
 Valdemar W. Setzer
 Manifesto Against the use of Computers in Elementary Education
- RT-MAP-8606 - (Agosto 1986) (105 pgs)
 Julio Michael Stern
 Fatoração L - U e Aplicações
- RT-MAP-8607 - (Agosto 1986) (73 pgs)
 Afonso Galvão Ferreira
 O Problema do Dobramento Optimal de PLAs
- RT-MAP-8703 - (Fevereiro 1987) (20 pgs)
 Imre Simon
 The Nondeterministic Complexity of a Finite Automaton
- RT-MAP-8704 - (Abril 1987) (8 pgs)
 Imre Simon
 Infinite Words and a Theorem of Hindman
- RT-MAP-8707 - (Agosto 1987) (36 pgs)
 Imre Simon
 Factorization Forests of Finite Height
- RT-MAP-8709 - (Março 1987) (31 pgs)
 Routo Terada
 Um Código Criptográfico para Segurança em Transmissão e Base de Dados
- RT-MAC-8801 - (Janeiro 1988) (21 pgs)
 Martin Groetschel, Yoshiko Wakabayashi
 Facets of the Clique Partitioning Polytope
- RT-MAC-8802 - (Fevereiro 1988) (52 pgs)
 Martin Groetschel, Yoshiko Wakabayashi
 A Cutting Plane Algorithm for a Clustering Problem
- RT-MAC-8803 - (Março 1988) (14 pgs)
 Martin Groetschel, Yoshiko Wakabayashi
 Composition of the Clique Partitioning Polytope
- RT-MAC-8804 - (Abril 1988) (14 pgs)
 Imre Simon
 Sequence Comparison: Some Theory and Some Practice
- RT-MAC-8805 - (maio 88) (14 pgs)
 Imre Simon
 Recognizable Sets with Multiplicities in the Tropical Semiring

RT-MAC-8806 - (junho 88) (40 pgs)

Valdemar W. Setzer; Ervino Marussi

LDT: Um Gerador Universal de Aplicações para Processamento
de Dados

RT-MAC-8807 - (agosto 88) (16 pgs)

Routo Terada

Probabilistic Analysis of Optimal Algorithms for Three NP-
Hard Problems