"CONTRIBUIÇÃO AO ESTUDO DE EFICIÊNCIA PARA PROGRAMAS FORTRAN BASEADO EM ESTIMATIVA DE TEMPO"

Simplício Lopes de Freitas

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.).

Aprovada por:

Prof. Pierre-Jean Lavelle
(presidente)

Prof. José Lucas M.Rangel Neto

Prof. Nelson Maculan Filho

RIO DE JANEIRO, RJ - BRASIL

outubro de 1976

DEDICATÓRIA

Ao meu pai

À minha mãe

À minha espôsa

Ao meu filho

AGRADECIMENTOS

Meus agradecimentos ao professor P. J. Lavelle, cuja orientação foi fundamental para esta tese; aos professores Lucas Rangel e Nelson Maculan pela ajuda e incentivo e por terem aceito em fazer parte da banca examinadora.

Estendo meus agradecimentos aos engenheiros W. Freire, P. Novaes e R. Fogaça, geofísicos da PETROBRÁS, e a S. Goldberg, a todos pelas sugestões, cooperação e estímulo.

SINOPSE

As instruções de máquina, que correspondem a cada comando executável em um programa Fortran, são determinadas através de análise da opção de listagem do código-objeto gerado pelo compilador Fortran. O tempo de execução do comando Fortran é então calculado, somando-se os tempos de execução das instruções que compõem cada comando. Esses elementos são inseridos na listagem de compilação do programa Fortran, dando ao programador subsídios para através de análises de custo e de tempo de cada comando, melhorar a eficiência da programação.

A grande vantagem do método é que ele se baseia nas características do equipamento que é usado, e no modo pelo qual o compilador Fortran gera o código-objeto, estando pois incorporados, a natureza dos dados, o tipo de operação a ser feito e todas as características descritas no programa.

ABSTRACT

The machine instructions corresponding to each executable statement in a Fortran program are determined by analyzing the output of the "LIST" option of the Fortran compiler. The execution time of the Fortran statement is then calculated by adding the execution times of these machine instructions. This and other information is inserted in the compiled listing of the Fortran program as an aid to the programmer in analyzing the cost and timing of each Fortran statement and improving program efficiency.

The major advantage of the method is that it is based on the characteristics of the machine being used, and on how the Fortran compiler generates the object code, which incorporates the nature of the data, the type of operations being performed and all the characteristics described in the program.

ÍNDICE

I. INTRODUÇÃO 1

II. REVISÃO DA LITERATURA 3

III. FUNDAMENTO TEÓRICO 4

III.1 BREVE ESBOÇO HISTÓRICO DOS COMPILADORES FORTRAN 4

III.2 DIFERENÇAS ENTRE OS COMPILADORES FORTRAN ANS 4

III.3 COMPILADORES OTIMIZADORES 5

IV. MATERIAL E MÉTODO 7

IV.1 TEMPO DE EXECUÇÃO DAS INTRUÇÕES 7

IV.2 LIMITAÇÕES DO CÓDIGO OBJETO 7

IV.3 LIMITAÇÕES DEVIDAS AO EQUIPAMENTO(HARDWARE) 9

IV.4 LIMITAÇÕES DEPENDENTES DA EXECUÇÃO 10

IV.5 DESENVOLVIMENTO DO PROGRAMA ANATEMP 11

IV.6 MÉTODO UTILIZADO NOS CÁLCULOS DOS TEMPOS 13

IV.7 OUTRAS INFORMAÇÕES ADICIONAIS GERADAS PELO PROGRAMA ANATEMP 15

IV.8 ESPECIFICAÇÕES DO PROGRAMA ANATEMP 16

IV.9 PROCEDIMENTOS DE UTILIZAÇÃO 18

IV.10 ESTIMATIVA DE CUSTO 18

V. RESULTADOS 20

V.1 TEMPOS FORNECIDOS PELO SISTEMA OPERACIONAL VERSUS TEMPOS CALCULADOS 20

V.2 CONSIDERAÇÕES SOBRE A MEDIÇÃO DE TEMPOS NOS SISTEMAS IBM/360 20

V.2.1 Testes efetuados em regime de multiprogramação 22

V.2.2 Testes efetuados sem o regime de multiprogramação 23

V.2.3 Outros testes 24

V.2.4 Considerações para computadores com outras características na arquitetura 27

VI. DISCUSSÃO 29

VI.1 OTIMIZAÇÕES DEPENDENTES DA MÁQUINA 29

VI.2 OTIMIZAÇÕES NA ESTRUTURA DO PROGRAMA 32

VI.3 OTIMIZAÇÕES EM SISTEMAS OPERACIONAIS VIRTUAIS 36

VI.4 OUTRAS OTIMIZAÇÕES 39

VII. CONCLUSÕES 41

VIII. REFERÊNCIAS BIBLIOGRÁFICAS 73

APÊNDICE 1 - PROGRAMA ANATEMP 42

APÊNDICE 2 - SUB-ROTINA MOVECO1 60

APÊNDICE 3 - SUB-ROTINA EBCDIC 63

APÊNDICE 4 - PROCEDIMENTOS CATALOGADOS 67

APÊNDICE 5 - SUB-ROTINA TIMEOT 70

I. INTRODUÇÃO

A eficiência de um programa de computação é função de vários fatores: - a correta formulação do problema, a seleção do algorítmo (uso de métodos numéricos), e a estrutura e técnica de programação. Na programação em linguagens de alto nível, a cuidadosa codificação dos comandos-fonte, de acordo com o compilador usado, assume também aspecto relevante.

As grandes diferenças existentes nos sistemas operacionais, nos modelos e tipos de máquinas e entre os muitos compiladores, impedem que os programadores disponham de regras fixas para a otimização de seus programas. Com o objetivo de superar essa dificuldade, procuramos investigar as informações contidas nas listagens do código-objeto, gerado pela grande maioria dos compiladores existentes, à procura de subsídios para a melhoria da eficiência das programações.

Dentre as várias linguagens de alto nível, escolhemos o Fortran, principalmente por sua característica de linguagem orientada para problemas matemáticos, onde os tempos de processamento são geralmente muito importantes. Além disso, o Fortran é a principal linguagem de aplicação do computador IBM 360/Mod.65, do Centro de Processamento de Dados Sísmicos da PETROBRÁS (CPDS), utilizado na elaboração deste trabalho.

Como resultado do presente estudo elaboramos um programa, denominado ANATEMP, que insere na listagem produzida pelo compilador Fortran várias informações, destacando-se en-

tre elas os tempos de execução de cada comando Fortran, com o que se arma o programador com novos elementos para melhoria da programação.

A grande vantagem do método é que ele se baseia nas características do equipamento que é usado, e no modo pelo qual o compilador Fortran gera o código-objeto, estando pois incorporadas, a natureza dos dados, o tipo de operação a ser feito e todas as características descritas no programa.

II. REVISÃO DA LITERATURA

A preocupação com a eficiência das programações tem levado inúmeros autores e fornecedores de software a discutirem, exemplificarem e criarem novas técnicas e métodos, que variam desde aqueles que se propõem a reduzir os tempos de execução e de UCP, até aqueles que visam a uma utilização melhor da memória. No entanto, a melhoria de eficiência pelo maior cuidado na codificação de programas-fonte são apresentadas de um modo generalizado e não, se aplicam indistintamente a qualquer compilador ou computador.

O manual para programadores Fortran da IBM[1] não faz qualquer referência à melhoria de eficiência para o compilador "G". Quanto ao compilador "H Extended", ela se limita a algumas observações rotineiras.

A estimativa e discussão dos tempos necessários para a execução de alguns algorítmos são abordados por KNUTH[2], para um computador hipotético e uma linguagem montadora.

Não se encontram na literatura sobre o assunto trabalhos referentes ao cálculo de tempo de execução de instruções, em linguagem de alto nível. Para a linguagem montadora merece destaque a fornecida pela Digital Equipment Corp., no cross-assembler do PDP-11 rodando no PDP-10.

III. FUNDAMENTO TEÓRICO

III.1 - BREVE ESBOÇO HISTÓRICO DOS COMPILADORES FORTRAN

Embora a linguagem Fortran venha perdendo a aceitação para as novas linguagens de alto nível tipo PL/I, ela ainda deverá ser utilizada por muito tempo devido (a) a larga popularização obtida desde o seu aparecimento em 1956/7; e (b) aos grandes investimentos que já se fizeram e estão sendo feitos com a sua utilização.

A linguagem Fortran tem sofrido, com o passar do tempo, grandes alterações. Em defesa dos usuários, o American National Standard Institute (ANSI) propôs, em 1966, duas versões padrões chamadas (1) Linguagem Fortran IV Básico e (2) Linguagem Fortran IV. Quaisquer programas escritos nos padrões ANSI podem ser compilados e executados em qualquer sistema tipo Fortran IV ANS.

Embora o software básico fornecido por grande parte dos fabricantes de computadores se apoie em compiladores Fortran não padronizados, a preferência generalizada dos usuários recai em compiladores tipo ANS.

III.2 - DIFERENÇAS ENTRE OS COMPILADORES FORTRAN ANS

Os compiladores Fortran ANS, de um modo geral, oferecem opções adicionais que variam de acordo com o fornece

dor. Sem levarmos em consideração o maior ou menor número de opções não contidas nos padrões ANSI, os compiladores se diferenciam nos seguintes pontos básicos:

a) Na velocidade de compilação, isto é, no tempo necessário para o compilador efetuar a tradução do programa-fonte em programa-objeto.
b) No tamanho da memória requerido pelo compilador. De uma maneira geral, os compiladores são programas que adquirem memória dinamicamente, à medida das necessidades. Em alguns deles, como o tipo "G", da IBM, 180K são suficientes para compilação de qualquer programa, mas já em outros como o tipo "H", do mesmo fornecedor, não há limites rígidos.
c) No tamanho do programa-objeto gerado. Um maior ou menor número de instruções e constantes podem ser utilizados no código-objeto gerado.
d) Na eficiência do programa-objeto gerado. O reconhecimento de expressões repetidas, a alocação de registradores e o uso de instruções mais rápidas, são os fatores mais importantes a serem considerados.

III.3 - COMPILADORES OTIMIZADORES

Compiladores especiais conhecidos como "compiladores otimizadores" visam a melhorar tanto a eficiência dos programas-objeto gerados, como a reduzir o tamanho desses pro

gramas. Para atingir tais propósitos, a velocidade da compilação é bem mais demorada e a necessidade de memória para o compilador é substancialmente maior. Acrescentem-se a isso os "pequenos" cuidados que o programador deve tomar, ao codificar determinados tipos de comandos (não descritas nas normas ANSI), sob a pena de não se obter os resultados esperados[1].

A melhoria de eficiência proporcionada por estes compiladores é apregoada pelos fornecedores e até recomendada, entre outros, por HEHL[3]. Infelizmente, estes compiladores não têm flexibilidade para otimizar determinados trechos de um programa. O processo de otimização é levado a efeito de uma maneira padronizada a todo o programa e para todos os programas, o que nem sempre é a melhor solução. "É muitas vezes difícil determinar a extensão e a natureza da otimização fornecida por um determinado compilador" como afirma HEHL[3], e a pesquisa que fizemos no manual de lógica para o Fortran IV-H, da IBM[4], não forneceu as respostas desejadas.

A falta de um documento definitivo, sobre a otimização feita por esses compiladores faz com que as técnicas de otimização realizadas pelo programador, tais como a eliminação do cálculo de expressões redundantes, sejam altamente aconselháveis na codificação de programas.

IV. MATERIAL E MÉTODO

A aplicação prática do método de otimização proposto neste trabalho foi dirigida para programas Fortran, compilados pelo "Compilador Fortran IV G Level 21", da IBM, utilizando-se o computador tipo IBM 360, model 65, com 1 MB de memória real, operando sob controle do sistem operacional "OS-MVT Release 21.8", instalado no Centro de Processamento de Dados Sísmicos do Departamento de Exploração e Produção da PETROBRÁS.

IV.1 - TEMPO DE EXECUÇÃO DAS INSTRUÇÕES

A determinação do tempo de execução das instruções procedeu-se de acordo com os tempos básicos e as diretrizes ditadas pelo fabricante da máquina para o modelo descrito[5].

No decorrer do presente estudo verificamos que muitas instruções do conjunto exsitente no computador IBM,360/65, não são utilizadas nos programas gerados pelo compilador Fortran G, motivo pelo qual não foram incluidas na tabela de instruções do programa que consta no Apêndice I, reduzindo-se, deste modo, o trabalho a ser feito para cálculo do tempo dessas instruções.

IV.2 - LIMITAÇÕES DO CÓDIGO-OBJETO

No código-objeto, gerado pelo compilador Fortran G, a única instrução do tipo SS (memória-a-memória) presente é a MVC (move). Deste modo, as complexas fórmulas de cálculo do tempo para este tipo de instruções não precisaram ser empregadas.

A instrução MVC no código-objeto gerado pelo compilador caracteriza-se por uma particularidade: só é utilizada no prólogo de subprogramas para substituir os argumentos "mudos" pelos "atuais". Por esta razão, ela é considerada de tempo fixo, determinada previamente, segundo uma das várias fórmulas existentes para o seu cálculo.

As instruções de STM (store multiple) e LM (load multiple) só são utilizadas nos prólogos e epílogos administrativos de programas ou subprogramas, com um total de duas STM e duas LM para cada seção de controle (control section). O limite de palavra-dupla (8 bytes nos sistemas IBM 360/370) necessário para o cálculo do tempo dessas instruções é tomado a partir do valor do deslocamento. É importante frisar que todo código-objeto, gerado pelo compilador, tem endereço inicial zero e que o programa de ligação (linkage-editor), ao combinar os diferentes módulos, sempre faz com que estes se iniciem a partir do limite de palavra-dupla, conseguindo, assim, manter em limite apropriado as variáveis definidas como palavra dupla, dentro das várias seções de controle.

A instrução BXLE (branch on index low or equal) só é utilizada na parte de controle de iterações. Isto conduz

a uma fácil identificação dos laços (loops), também existentes implicitamente em comandos de entrada/saída. No cálculo do tempo de execução desta instrução consideramos o desvio uma ocorrência constante (a ausência do desvio provoca um êrro de apenas 1,4% no tempo de instrução).

IV.3 - LIMITAÇÕES DEVIDO AO EQUIPAMENTO (HARDWARE)

A IBM, ao fornecer as fórmulas e os tempos das instruções, indica as premissas assumidas e as considerações básicas inerentes a cada modelo.

Para o modelo 360/65[5], tendo-se em vista o código-objeto, gerado pelo compilador Fortran G, aplicam-se as seguintes limitações relacionadas com os equipamentos:

1. O tempo fornecido é a soma do tempo de decodificação e de execução da instrução.
2. As instruções podem começar em limite de meia-palavra (2 bytes) ou palavra (4 bytes), com igual probabilidade. Exceções: todas as instruções do tipo SS e as instruções STM e LM.
3. Interrupções, caso ocorram, não são incluídas nos tempos dados.
4. O tempo necessário para a indexação por um registrador-base já está incluído no tempo fornecido. Para instruções que podem ser duplamente indexadas, um tempo adicional de

ve ser incluído (este requisito é satisfeito no programa desenvolvido).

5. Em todas as operações aritméticas, o sinal dos operandos (negativo e positivo) tem igual probabilidade de ocorrência.
6. O tempo das instruções de ponto flutuante depende (a) do número de digitos hexadecimais que são pré e pós-movidos e (b) do número de vezes que o resultado é recomplementado. Os tempos fornecidos são uma média destas variáveis.

IV.4 - LIMITAÇÕES DEPENDENTES DA EXECUÇÃO

Limitações adicionais se fazem presentes quando a fórmula de cálculo do tempo de determinada instrução inclui fatores só determinados em tempo de execução. Para estas instruções, foram tomados os seguintes pressupostos:

1. As interrupções que podem ocorrer nas operações de divisão de ponto-fixo (aumentando o tempo de execução destas em cerca de 1,76%) não são consideradas.
2. Em todas as instruções de desvio, este é levado em consideração, exceto se, no código-objeto gerado, houver "máscara zero" nas instruções tipo RX e RR. Ocorrerá também a mesma exceção no tipo RR, caso o segundo operando seja igual a zero.

IV.5 - DESENVOLVIMENTO DO PROGRAMA ANTEMP

Como aplicação prática, desenvolvemos o programa ANATEMP, que após a etapa de compilação, e de acôrdo com o tempo básico de UCP e a fórmula associada para cada instrução, calcula o tempo de execução para cada comando Fortran e demais informações.

Todas as etapas abaixo descritas são ilustradas na Figura IV.5.

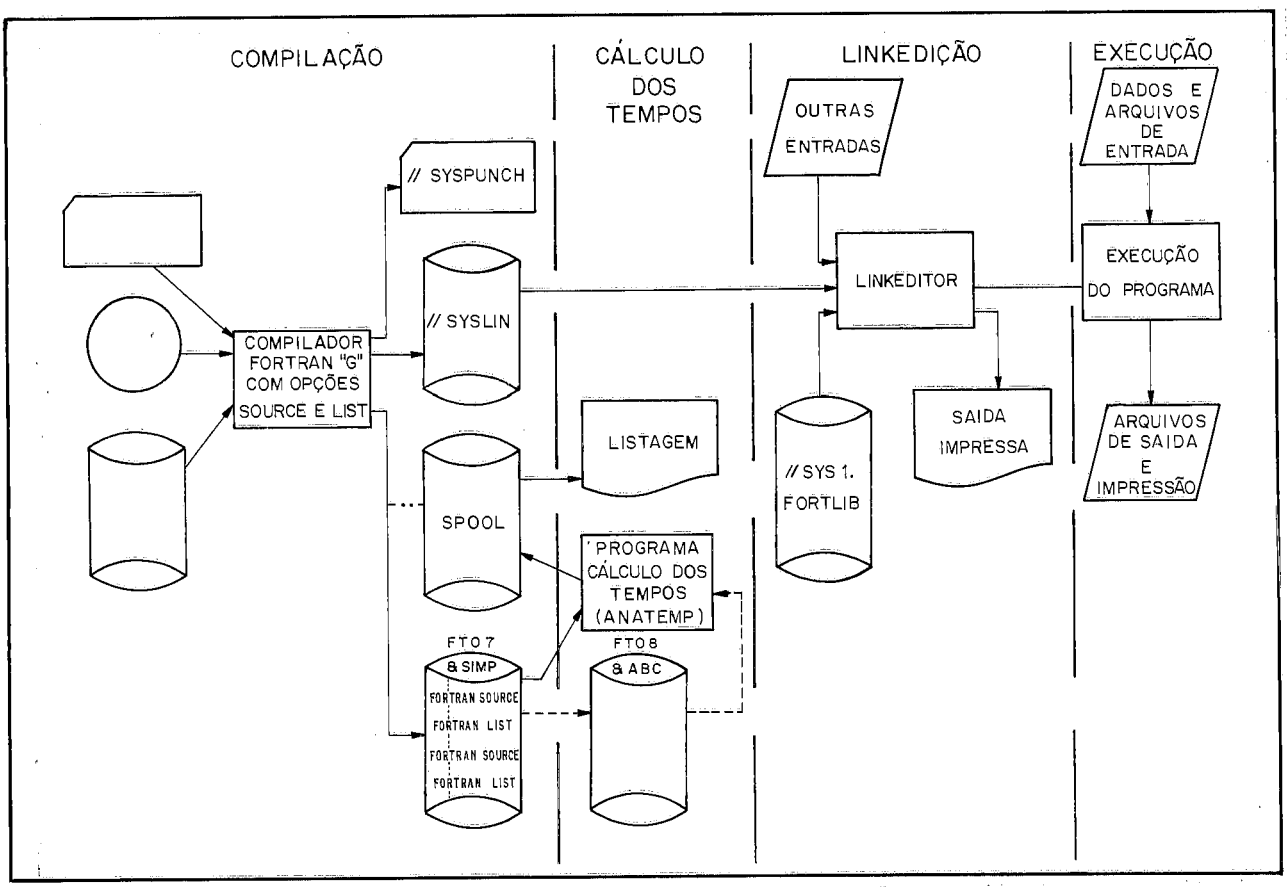

FIGURA IV.5 FLUXO GERAL DE UM PROGRAMA FORTRAN COM A INCLUSÃO DO PROGRAMA ANATEMP

Compilação - A compilação do programa é feita, no primeiro passo, com as opções de listagem do programa-fonte (source) e listagem do programa-objeto (list), ambas obrigatórias. A saída impressa(//SYSPRINT) desta etapa é desviada do spool para uma área de disco (DSN=&SIMP). Compilações sucessivas ou em grupo (batch) podem ser feitas e cada unidade é tratada separadamente, no passo seguinte.

Programa de Cálculo dos Tempos (ANATEMP) - Resumidamente, são as seguintes as etapas executadas.

1. A parte source é lida de DSN=&SIMP (FT07F001) e levada para a memória. Se esta área for insuficiente para conter todas as informações, utiliza-se o espaço em disco, definido em DSN=&ABC.

   Se a opção list for encontrada e o DSN=&ABC tiver sido utilizado, este é então, reposicionado para o início.
2. As opções source e list são, neste ponto, analisadas simultaneamente, procedendo-se a todos os cálculos e efetuando-se a impressão dos resultados.
3. Encontrado o final do programa ou do subprograma, faz-se um teste para a verificação da existência de mais subprogramas; em caso positivo, os passos 1 e 2 são repetidos.

"Linkedição" e Execução - Estas etapas podem ser realizadas normalmente e suas entradas e saídas não são trabalhadas pela eta

pa anterior.

IV.6 - MÉTODO UTILIZADO NOS CÁLCULOS DOS TEMPOS

O tempo calculado pelo programa ANATEMP e definido como LINK tem significado diferente para a análise de programas e subprogramas. Para os primeiros, é o tempo gasto com a ligação necessária ao programa de controle (supervisor). Para os subprogramas, é o total do prólogo, isto é, a soma dos tempos necessários para salvar conteúdo dos registradores, "inicialização" e substituição dos argumentos mudos pelos atuais.

Tempo de Execução de um Comando (Col.7) - É o tempo total esperado para a execução de todas as instruções que compõem o comando, inclusive as instruções que compõem os subprogramas abertos, porventura utilizados. Caso o comando faça alguma referência externa (subprograma fechados), o tempo de execução representa apenas o valor gasto dentro de programa ou subprograma. Os comandos que têm referências externas são detectados e esta informação bem como o número de chamadas é anexada às informações fornecidas ao programador, nas colunas 2 e 4.

Tempo mínimo de execução de um comando (Col.6) - Para os comandos onde é possível haver mais de um tempo de execução, este valor representa o menor tempo possível. A informação é particularmente útil em comandos compostos, como no exemplo (IV.6.

1), onde a expressão AI = AI/AB só será calculada se a primeira expressão for verdadeira.

Exemplo (IV.6.1)

```
IF (A.GT.B.AND.G.LE.H)  AI = AI/AB
```

Cálculo do Tempo de "DO loops" - Para iterações feitas através de comandos "DO", calculam-se os seguintes valores: a) tempo para uma iteração (col.8) é apresentado entre parênteses, na saída do programa, indicando que este é o valor que deve ser multiplicado para se obter o valor total de "n" iterações, exceto a primeira; b) tempo da primeira iteração: ao valor acima obtido, na letra a, deve ser adicionado o número que se lhe segue (col.9). O valor preciso para "n" iterações, deverá ser calculado pela fórmula (IV.6.2), embora para fins práticos a fórmula (IV.6.3) possa ser utilizada.

Fórmula IV.6.2

(valor entre parênteses + valor seguinte) +((n-1) x valor entre parênteses).

Fórmula IV.6.3

(n x valor entre parênteses)

Três pilhas (stack) são usadas pelo programa, no cálculo dos tempos de laços (DO loops), prevendo-se assim a possibilidade de cálculo dos tempos de laços contidos em um

"ninho" de DO, de vez que estes podem ser tratados de modo LIFO (last-in-first-out):

IV.7 - OUTRAS INFORMAÇÕES ADICIONAIS GERADAS PELO PROGRAMA ANATEMP

Tendo-se verificado que os espaços em branco, existentes na listagem do programa fonte e não utilizáveis pelo programador são superiores às necessidades primárias do programa, adicionamos outras informações suplementares, consideradas úteis para a depuração de programas.

Todos os comandos que podem fazer desvios (exceto os desvios feitos por referências externas, já descritos) são reconhecidos pela presença da letra "B", na coluna 1.

Todos os comandos que finalizam um laço "DO" ou que contêm uma iteração implícita, como no exemplo (IV.7.1), são determinados e apresentados pela letra "L", na coluna 3.

Exemplo (IV.7.1)

```
READ (1,8) (A(I),   I = 1,J)
```

A locação relativa a zero, de onde um comando tem início, é também colocada junto a este (col.5). A informação dá também ao programador duas outras opções sem a necessidade de consulta à opção LIST. É uma excelente arma para a depuração, pois a PSW (program status word) gerada, menos o

ponto de entrada (entry-point), muitas vezes dá a localização do comando em êrro. Torna-se também simples saber quantos bytes são utilizados em cada comando, bastando, para isso, efetuar a diferença entre o comando desejado e o próximo.

Um resumo contendo as instruções e o número de ocorrências destas no código-objeto, gerado pelo compilador, é também apresentado na parte final da listagem, juntamente com a soma dos tempos de todos os comandos do programa. Este tempo não representa o tempo esperado de execução, pois geralmente o programa faz iterações em determinados trechos; mas ele dá uma idéia que pode ser útil, quando se deseja fazer comparações de um mesmo programa que tenha sofrido pequenas alterações.

IV.8 - ESPECIFICAÇÕES DO PROGRAMA ANATEMP

A codificação do programa ANATEMP (Apêndice 1), na linguagem Fortran, está comentada no próprio texto, permitindo aos programadores Fortran sua fácil adaptação, se necessário. Entretanto tivemos de fazer uso de duas sub-rotinas externas em linguagem montadora: a sub-rotina MOVECO1 e a sub-rotina EBCDIC. Ambas são de "uso geral" e não deviam faltar entre as sub-rotinas supridas pela "livraria de subprogramas Fortran".

A sub-rotina MOVECO1, tem duas entradas: a primeira, MOVEC ou MVC, através de uma lista de 5 parâmetros, ser

ve para mover "n" ($256 \leq n \geq 1$) caracteres de uma locação de memória para outra; a segunda MOVE, faz a transferência de 132 bytes de uma locação de memória para outra usando apenas dois parâmetros. O tamanho da sub-rotina MOVECO1 é de 104 bytes. (Apêndice 2).

No Apêndice 3, está a sub-rotina EBCDIC, que através de duas entradas, PAREBC e DEBC, faz conversões do tipo "A", "E", "I", "F", "D" de números para EBCDIC e vice versa. O tamanho desta sub-rotina é de 340 bytes.

Para facilitar a estrutura do programa, pois entre outras simplificações, não haveria a necessidade do uso de pilhas no cálculo dos tempos das iterações, tentamos utilizar o método de acesso direto a discos. Constatamos,mais uma vez, que o uso desta técnica não é recomendável onde o método sequencial pode ser aplicado. Os tempos de UCP e de execução, principalmente este último, usando-se a técnica de acesso direto, tornariam o uso rotineiro do programa anti-econômico. A utilização do método sequencial permite que se faça a blocagem dos registros lógicos a serem trabalhados, evitando-se, assim, excessivas chamadas às rotinas do supervisor de E/S. Para os discos modelo 2314/2313, ligados ao computador 360/65 do CPDS, o tamanho do bloco para DSN=&SIMP foi de 7200 bytes e 7260 bytes para o DSN=&ABC. Em instalações em que outros modelos de disco seja utilizados, esses tamanhos deverão ser alterados, para melhor eficiência.

Na codificação do programa ANATEMP, também

prevemos a possibilidade de se analisar programas dentro do formato Fortran "livre". Portanto, nenhuma restrição impomos à utilização deste programa, exceto aquelas que também não são admitidas pelo compilador. Para efeito de eficiência, as expressões do tipo DO10I=1, que não representam um comando "DO" devem ser evitadas.

IV.9 - PROCEDIMENTOS DE UTILIZAÇÃO

Uma vez implantados os procedimentos desenvolvidos no Apêndice 4, a utilização do programa ANATEMP, em sistemas operacionais "OS", pode ser feita de modo e maneira idêntica aos procedimentos catalogados para Fortran G, exceto pela substituição da letra "G" por "T", no cartão EXEC, conforme mostra o exemplo IV.9.1.

Exemplo IV.9.1

```
//PASSO1  EXEC FORTTCLG ao invés de
//PASSO1  EXEC FORTGCLG
```

IV.10 - ESTIMATIVA DE CUSTO

Pode-se proceder à avaliação do custo de utilização do programa de análise dos tempos (ANATEMP), de modo resumido, a partir da comparação entre os tempos de execução e UCP gastos pelo compilador FORTRAN e o programa ANATEMP pa-

ra os exemplos mostrados na Tabela IV.10.1. Estes valores foram obtidos através dos relatórios de SMF[6] também disponíveis no CPDS.

TABELA IV.10.1 - COMPARAÇÃO ENTRE TEMPOS UCP E EXECUÇÃO GASTOS PELO COMPILADOR FORTRAN "G" E O PROGRAMA ANATEMP

	PROGRAMAS TESTADOS				TOTAL
	PROG.1	PROG.2	PROG.3	PROG.4	
Número total de Cartões de entrada	325	626	766	1475	3192
Número de Comandos-fonte Fortran	168	427	393	960	1948
Tempo UCP para compilador Fortran G(seg) ...	6,23	18,42	17,43	53,58	95,75
Tempo UCP para ANATEMP (seg)	5,60	13,32	10,73	30,98	60,63
Tempo Execução para Compilador Fortran G(min).	0,6	0,8	1,0	2,1	4,5
Tempo Execução ANATEMP (min)	0,5	0,8	0,4	1,8	3,5

Da Tabela IV.10.1, pode-se inferir que o custo do programa ANATEMP, comparado com o custo de compilação do mesmo programa, é consideravalmente menor, já que é em média 25% inferior em tempo de execução e 50% menor em termos de tempo de UCP.

V. RESULTADOS

V.1 - TEMPOS FORNECIDOS PELO SISTEMA OPERACIONAL VERSUS TEMPOS CALCULADOS

A eficácia do método proposto neste trabalho pode ser verificada, praticamente, através de testes comparativos entre os tempos calculados pelo programa e os fornecidos pelo sistema.

Antes da apresentação dos resultados, discutiremos, a seguir, as considerações aplicáveis na análise dos mesmos.

V.2 - CONSIDERAÇÕES SOBRE A MEDIÇÃO DE TEMPOS NOS SISTEMAS IBM 360

As seguintes ponderações são fornecidas pela IBM, no manual OS-SMF[6].

O tempo de processamento de um serviço (job) é aquele compreendido entre o início e o término de execução de um programa-problema. Inclui o tempo usado pelo programa de controle, em algumas de suas fases, excluindo, entretanto, os tempos usados pelo scheduler*, e pelos programas de reader** e writer***.

O tempo de processamento não pode ser considerado constante sempre que o mesmo passo (step) ou job for executado. Um ou mais dos seguintes fatores, podem ser causadores destas variações: interrupções pendentes, arquitetura da UCP, tomada de ciclo (cycle stealing) de canais e restauração (retries), em programas de canal de mudança de tarefas (tasks).

Observações adicionais referentes aos serviços de tempo do computador IBM 360 são também descritas pela IBM[7]. A mais importante destas é a resolução do relógio(timer), cuja fórmula para atualização é igual ao inverso da frequência da linha. Para uma linha de 60 Hz a atualização só é feita a

---

* Conjunto de programas do sistema operacional que escalonam os serviços.

** Programa do sistema operacional que lê os jobs dos dispositivos de entrada e os coloca nos dispositivos de acesso direto.

*** Programa do sistema operacional que grava a saída contida em dispositivo de acesso direto para os dispositivos de saída.

cada 16,666 milisegundos, e um êrro de 1,7%, nos valores apresentados, pode então ser causado pela variação de apenas um hertz na linha.

V.2.1 - Testes efetuados em regime de multiprogramação

Os resultados relativos aos testes em regime de multiprogramação são apresentados na Tabela V.2.1.1. e correspondem à execução sucessiva, em regime de multiprogramação, de um mesmo programa (teste A a D) constituído de um único laço "DO", que continha apenas operações de ponto fixo.

TABALE V.2.1.1 - TEMPOS PARA UM MESMO PROGRAMA EXECUTADOS EM REGIME DE MULTIPROGRAMAÇÃO

	TESTES				
	A	B	C	D	E
a) Tempo fornecido pelo sistema (seg).....	9,85	11,92	9,95	10,73	111,48
b) Tempo calculado pelo ANATEMP (seg).....	10,21	10,21	10,21	10,21	102,10
Diferença % (b/a).....	-3,52	16,74	-2,54	5,09	9,29

Os valores do teste E, na Tabela V.2.1.1, foram obtidos através da execução do mesmo programa com maior número de iterações. Do exame da tabale V.2.1.1 verifica-se que em regime de multiprogramação os tempos fornecidos pelo sistema podem sofrer variações significativas.

V.2.2 - Testes efetuados sem o regime de multiprogramação

A Tabela V.2.2.1 mostra os resultados obtidos com a execução do mesmo programa, diversas vezes, mas de cada vez, sempre, com o uso exclusivo do sistema, evitando-se,assim, os desvios provocados pela concorrência de vários programas. Verifica-se, desta feita, que os tempos fornecidos pelo sistema sofrem muito menor variação, obtendo-se portanto, valores muito próximos aos calculados pelo ANATEMP.

TABELA V.2.2.1 - TEMPOS DE EXECUÇÃO COM CONTROLE EXCLUSIVO DO COMPUTADOR

	TESTES		
	A	B	C
a) Tempo fornecido pelo sistema (seg)	9,88	9,68	97,80
b) Tempo calculado pelo ANATEMP (seg)	10,21	10,21	102,10
Diferença % (b/a)	-3,23	-5,18	-4,11

V.2.3 - Outros Testes

Procedemos a outros testes face a possibilidade de uso, através do sistema operacional do CPDS, de macro-instruções assembler , que permitem "inicialização", teste e cancelamento de intervalos de tempo, embora esses testes estejam sujeitos também às limitações já apontadas anteriormente.

Desenvolvemos a sub-rotina TIME01 (Apêndice 5) em linguagem montadora. A entrada TIME permite medir os tempos de UCP entre os diversos pontos de um programa que inclui, neste caso, os tempos de todos os subprogramas externos, porventura existentes no trecho, como, por exemplo, rotinas de E/S.

A série de testes elaborados consistiu em chamadas à sub-rotina TIME01, em diversos pontos de programas compostos de diversas iterações, com diferentes tipos de operações de ponto fixo e ponto-flutuante, tendo-se evitado possíveis referências externas.

A Tabela V.2.3.1 mostra os resultados obtidos, quando o teste foi executado com o uso exclusivo do computador e a Tabela V.2.3.2, os resultados obtidos no regime de multiprogramação.

Uma vez mais, verifica-se que os valores obtidos com o uso exclusivo do sistema são os mais próximos dos obtidos pelo ANATEMP, e que os resultados obtidos pela sub-rotina TIME01 são de ótima precisão quando comparados aos fornecidos pelos sistema.

Tabela V.2.3.1 - RESULTADOS OBTIDOS ATRAVÉS DA SUB-ROTINA TIMEO1 COM O USO EXCLUSIVO DO SISTEMA

TESTE	1 TEMPO TOTAL PARA O PROGRAMA FORNECIDO PELO SISTEMA (seg)	2 TEMPOS CALCULADOS PARA O TRECHO PELO ANATEMP (seg)	3 TEMPOS FORNECIDOS PELA TIMEO1 (seg)	DIFERENÇA % (1)/(2)	DIFERENÇA % (3)/(2)	DIFERENÇA % (1)/(3)
A		11,41 12,44 14,87 0,00	12,20 12,96 16,67 0,00			
	42,23	38,72	41,83	9,05	8,03	0,09
B		51,94 56,31	51,41 58,30			
	110,10	108,25	109,71	1,71	1,34	0,03

TABELA V.2.3.2 - RESULTADOS OBTIDOS ATRAVÉS DA SUB-ROTINA TIMEO1 EM REGIME DE MULTIPROGRAMAÇÃO

	1	2	3			
TESTE	TEMPO TOTAL PARA O PROGRAMA FORNECIDO PELO SISTEMA (seg)	TEMPOS CALCULADOS PARA O TRECHO PELO ANATEMP (seg)	TEMPO FORNECIDOS PELA TIMEO1 (seg)	DIFERENÇA % (1)/(2)	DIFERENÇA % (3)/(2)	DIFERENÇA % (1)/(3)
A		11,41	12,25			
		12,44	13,57			
		14,87	17,82			
		0,00	0,00			
	43,95	38,72	43,64	13,5	12,7	0,07
B		51,94	51,87			
		56,31	58,85			
	111,62	108,25	110,72	3,1	2,3	0,08

V.2.4 - Considerações para computadores com outras características de arquitetura

Para o modelo 360/65, como já foi visto, os resultados fornecidos pelo programa refletem, com boa precisão, os tempos reais de execução e isto também é esperado para todos os computadores de arquitetura semelhante.

Para os modelos em que a CPU pode ter um cache (high speed buffer),onde o ciclo de memória do sistema pode ser reduzido a uma fração do ciclo de memória do processador, grandes variações podem ocorrer. Os tempos fornecidos pelo fabricante para máquinas com esta característica, como o modelo 370/165, da IBM, são fornecidos em relação às referências feitas pelo cache. Se um bloco não está no cache, o tempo poderá sofrer, em alguns casos, um aumento superior a 200%. Devido também à complexidade dos circuitos empregados nestes modelos, os tempos médios de execução de instruções, consideradas a partir do cache, podem ainda variar em até 28,5%, conforme a IBM[8].

Diferenças também significativas podem ocorrer nos modelos, onde o DAT (Dynamic Address Translation Mode) é instalado e o programa executado no esquema "virtual".

Para os computadores com essa arquitetura, o método utilizado no ANATEMP representa pois, a situação mais favorável, ou seja, instruções já no cache e sem o uso do DAT. Para levarmos em consideração os efeitos desses dispositivos, seria necessário que eles fossem simulados pelo programa, o que

fugiria ao escopo do presente trabalho, tendo em vista que o computador IBM 360/65 não dispõe de cache ou de DAT.

Podemos, no entanto, considerar úteis os cálculos de tempo sem a simulação destes fatores, nos computadores com os dispositivos acima mencionados, uma vez que o valor relativo dos tempos, calculados pelo ANATEMP, para os diferentes comandos executáveis, é correto.

VI. DISCUSSÃO

Nosso propósito, neste capítulo, é sumarizar alguns aspectos e técnicas que podem ser utilizados na codificação de programas de computação, visando a aumentar-lhes a eficiência.

O advento de sistemas operacionais que trabalham com o conceito de memória virtual fez com que as "regras" de tamanho e ocupação de memória e de eficiência fosse reconsideradas em relação aos sistemas operacionais convencionais. Por este motivo, discutimos, separadamente, algumas práticas de programação a serem empregadas nestes diferentes sistemas operacionais.

VI.1 - OTIMIZAÇÕES DEPENDENTES DA MÁQUINA

A Tabela VI.1 mostra os tempos de execução de algumas instruções para diversos modelos IBM. Deve-se observar que para os modelos 370/145,158 e 370/165 os tempos indicados são sempre os tempos "médios mínimos", fornecidos pelo fabricante, definidos como a situação mais favorável (as instruções a serem executadas já estando no cache e sem uso do DAT).

A Tabela VI.2 mostra os principais componentes dos computadores referenciados na Tabela VI.1, bem como o custo aproximado da execução das instruções da referida tabela. Advertimos que esta é uma análise simplificada em que apenas alguns fatores são considerados, fatores que não seriam os únicos a serem levados em conta na escolha de um computador.

TABELA VI.1 TEMPO DE EXECUÇÃO DE INSTRUÇÕES (EM MICROSEGUNDOS) EM DIVERSOS MODELOS IBM

INSTRUÇÕES	M_{NEM}	360/40	360/44	370/145	360/65	370/158	370/165
Load Halfword	LH	10,63	2,25	2,295	1,40	0,933	0,16
Load	L	11,88	2,25	1,688	1,20	0,588	0,16
Load (Short)	LE	11,88	2,25	1,688	1,20	0,703	0,16
Lod (Long)	LD	16,88	4,25	2,633	1,40	0,703	0,16
Store Halfword	STH	10,00	2,50	1,498	1,73	0,875	0,32
Store	ST	12,50	2,50	1,497	0,93	0,645	0,32
Store (Short).....	STE	12,50	2,50	1,497	0,93	0,875	0,32
Store (Long)	STD	17,50	4,50	3,386	0,93	0,875	0,32
Add Halfword	AH	10,63	2,25	2,949	1,80	1,163	0,16
Add	A	11,88	2,25	2,385	1,40	0,933	0,08
Add Norm. Short	AE	18,85	4,56	6,737	2,43	2,400	0,38
Add Norm. Long	AD	39,70	7,53	8,265	2,45	2,500	0,30
Subtract	S	11,88	2,25	2,340	1,40	0,933	0,16
Subtract Halfword ...	SH	10,63	2,25	2,949	1,80	1,163	0,16
Subtract Norm. Short	SE	20,62	4,56	7,041	2,43	2,400	0,38
Subtract Norm.Long..	SD	41,30	7,53	8,570	2,45	2,500	0,30
Multiply Halfword ..	MH	45,00	10,72	10,508	5,00	1,416	0,80
Multiply	M	86,40	16,89	20,077	4,80	1,991	0,78
Multiply Short	ME	80,60	14,81	16,795	4,40	2,100	1,15
Multiply Long	MD	259,40	62,64	45,673	7,60	3,900	1,87
Divide	D	196,70	29,00	34,771	8,70	9,903	1,96
Divide Short	DE	141,08	24,00	28,702	7,30	8,900	1,64
Divide Long	DD	480,25	125,25	89,565	14,10	23,300	2,65

TABELA VI.2 CUSTO APROXIMADO DA EXECUÇÃO DAS INSTRUÇÕES DA TABELA VI.1 EM DIVERSOS COMPUTADORES IBM.

	360/40	360/44	370/145	360/65	370/158	370/165
Processador	2040H	2044H00	3145J02	2065J00	3158J00	3165J00
Dispositivos obrigatórios incluídos no processador	2ch. sel. 1ch. mpx.	2ch.high speed byte MPX	VS		VS 2ch.blk.mpx. 1ch. mpx	não VS
Memória	256 K	256 K	1 MB	1 MB	1 MB	1 MB
Aluguel mensal em cruzeiros *	185.500	176.500	314.500	775.200	540.600	778.600
Soma dos tempos de todas as instruções da Tabela VI. 1 (em microsegundos)	1.559,09	339,49	303,51	77,78	71,70	14,69
Custo de execução das instruções ** (10^{-6} Cr$)	111,57	23,11	36,82	23,26	14,95	4,4

* Aluguel mensal refere-se somente ao equipamento listado, sem considerar-se os "encargos iniciais" (basicamente IPI, ICM, Imposto de Importação, Frete e Seguro), que montam em aproximadamente de 10 a 12 vezes o valor do aluguel mensal (e pagos de uma só vez) para equipamentos novos (preços em junho de 1976).

** O custo de execução foi obtido através do produto do aluguel mensal vezes o tempo total para a execução das instruções da Tabela VI.1, dividido pelo número de microsegundos de um mês (2592×10^9).

Algumas otimizações podem ser feitas a partir do conhecimento do tempo de execução das instruções e do modo pelo qual o código-objeto é gerado pelo compilador,conforme se segue:

1. Para o modelo 360/40 todas as operações realizadas com variáveis e conjuntos especificados, como INTEGER*2(2 bytes) são significativamente mais rápidas que as mesmas operações realizadas em INTEGER*4(4 bytes). O oposto é correto para o modelo 360/65 e, de um modo geral, para todos os outros modelos, conforme se vê na Tabela VI.1.
2. Deve-se dar preferência às instruções de "Adição"(Add) uma vez que elas são, algumas vezes mais rápidas que as de "multiplicação"(Multiply), substituindo-se expressões do tipo (2*I ou 3*J) por adições sucessivas, ex. (I + I) ou (J + J + J).
3. De modo análogo, as expressões do tipo (REAL/2.0) serão otimizadas ao usar-se (REAL*0.5), de vez que a multiplicação é mais rápida do que a divisão.
4. As operações realizadas sobre inteiros são mais rápidas do que as de ponto flutuante e estas, mais rápidas que as de dupla precisão. Deste modo, sempre que possível, deve-se optar pela ordem INTEGER*4 ou INTEGER*2, REAL*4 e REAL*8.

VI.2 - OTIMIZAÇÕES NA ESTRUTURA DO PROGRAMA

A característica mais importante da estrutura do módulo de carga de programas Fortran é que as variáveis, vetores e matrizes estão contidas na mesma seção de controle(CSECT) que o código de execução e, portanto, não podem ser tratados separadamente pelo programa de ligação(linkage-editor).

Há uma seção de controle para o programa principal (MAIN), uma para cada subprograma, uma para a área de "comum em branco" (BLANK COMMON) e uma para cada área de comum (COMMON) com nome. A Figura VI.2 mostra a forma geral da estrutura do módulo de carga de um programa Fortran, que tem todas as seções de controle descritas.

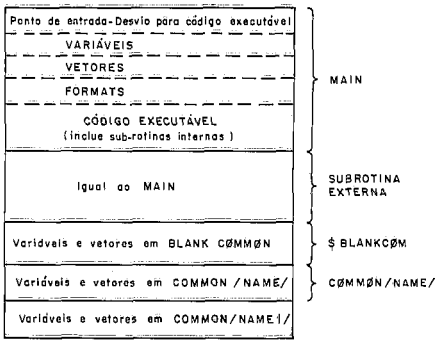

FIG. VI.2-ESTRUTURA DE MÓDULO DE CARGA DE UM PROGRAMA FORTRAN

São as seguintes as sugestões aplicáveis para se aumentar a eficiência do programa através de melhor alocação de registradores no código-objeto gerado e, portanto, pela redução do número de instruções "L" (Load):

1. Utilizar técnicas ou algorítmos que usem as menores áreas de trabalho.
2. Declarar os vetores e matrizes na ordem em que se espera referenciá-los mais frequentemente. Isto implica em declarar o dado tão perto quanto possível de outro dado que será usado ao mesmo tempo. No exemplo VI.2.1, a declaração em (a) é melhor que a usada em (b), se o programa fizer o multiplicação das matrizes A e B e o resultado for colocado na matriz C.

Exemplo VI.2.1

(a) DIMENSION A(10,10), B(10,10), C(10,10), D(10,10), E(5000), J(500)

(b) DIMENSION A(10,10), E(5000), J(500), B(10,10), D(10,10), C(10,10)

3. Declarar as áreas na ordem de crescimento de posição a serem ocupadas, desde que o item (2) anterior tenha precedência.

   No exemplo VI.2.2 o comando (c) é preferível ao (d).

Exemplo VI.2.2

(c) DIMENSION A(10), B(10), C(12), D(20), E(5000)

(d) DIMENSION E(5000), A(10), D(20), C(12), B(10)

De um modo geral, considera-se que programas constituídos de módulos extensos tendem a ficar complexos e de difícil manutenção. A divisão de um "grande" programa em subprogramas é estruturalmente útil para os programadores.

Os subprogramas dividem-se em dois grupos: abertos ou em linha e fechados ou externos.

Subprogramas abertos são aqueles inseridos no programa que os chama, sempre que forem referenciados. Este processo é utilizado pelo compiladores Fortran G e H para realizar funções que requerem poucas instruções, por ex., MOD, IFIX, MIN.Também fazem parte dos subprogramas abertos as sub-rotinas implícita ou explicitamente chamadas, para a conversão de dados, por ex., FLOAT, IFIX. Este tipo de sub-rotina leva à diminuição do tempo de UCP e ao aumento de memória utilizável. Conforme visto na Figura VI.2, os subprogramas a que nos referimos ficam fazendo parte da seção de controle ao qual foi incorporado.

É sempre custoso, em termos de tempo, fazer referências externas. Quando um subprograma externo é chamado (implícita ou explicitamente), o tempo necessário para fazer a ligação entre este e o que chamou é constituído de três partes:

1) tempo de ligação propriamente dito, ou seja, o tempo necessário para executar as instruções de desvio para o subprograma;
2) tempo de prólogo, que é a "inicialização" do subprograma;
3) tempo de epílogo, que é a finalização do subprograma e consiste em restaurar os registradores, possivelmente retornar os

resultados e devolver o controle ao programa de onde foi chamado.

São subprogramas fechados a maioria das funções providas pelo Fortran IV e frequentemente utilizadas pelo programador, tais como SIN, SQRT, e também as referenciadas pelos comandos de Entrada/Saída (READ, WRITE, etc).

Aplicam-se aos subprogramas, escritos pelo programador, as seguintes observações:

1) Quando, do ponto de vista estrutural, for aconselhável utilizar subprogramas externos, o melhor é procurar minimizar o número de parâmetros ou colocá-los em uma só área de COMMON.
2) Se um subprograma é de uso específico de determinado programa, e tendo em vista o tamanho e o número de referências a ele feitos, deve-se preferentemente definí-lo como interno, conforme mostra o exemplo VI.2.3. O tempo gasto para o desvio interno é irrisório, se comparado a uma chamada externa.

Exemplo VI.2.3

```
    ASSIGN 100  TO  IVOLTA
    GO  TO  300
100 .....
    .
    .
    .
    ASSIGN 200  TO  IVOLTA
    GO  TO  300
200 .....
    .
    .
    .
    COMANDOS DO SUBPROGRAMA INTERNO
C
300 .....
    .
    .
    .
    GO TO IVOLTA, (100,  200, ...)
```

VI.3 - OTIMIZAÇÕES EM SISTEMAS OPERACIONAIS VIRTUAIS

Os sistemas virtuais, aqui discutidos, são aqueles implementados pela IBM, nos computadores da linha 370. Nestes sistemas, o espaço virtual e real são divididos em partes

fixas, conhecidas como páginas. Em um momento qualquer há, certo número de páginas que pertencem a um programa qualquer e estão nas páginas da memória real. Quando se faz referência a um endereço que não está na memória real,ocorre uma interrupção chamada "exceção de página" (page fault), na qual o sistema operacional deverá trazer a página solicitada para a memória real, num processo denominado paginação.

O número de páginas virtuais é, normalmente, bem maior que o número de páginas reais. Em certos programas, as necessidades de paginação podem tornar-se tão altas que os recursos do sistema são usados mais para o processo de paginação, do que para a execução destes programas, gerando-se então, uma grande ineficiência no sistema, conhecida por thrashing.

Alguns programas, quando executados em sistemas virtuais, tendem a causar excessiva paginação. Nesses casos, técnicas especiais de programação devem ser empregadas para minimizar esse efeito. Infelizmente, estas técnicas são, muitas vezes, conflitantes com as normalmente aplicadas em programas a serem executados em sistemas operacionais convencionais e com técnica de programação estruturada.

Um dos melhores métodos para a redução da paginação excessiva de um programa é a redução do tamanho total deste. Neste caso, o programa terá menor número de páginas que, por sua vez, gerarão menor número de page faults. Evidentemente, esta prática nem sempre é seguida, pois o programador, de um modo geral, é apenas informado de que possui uma quantidade

"ilimitada" de memória, adquirindo práticas extravagantes no dimensionamento das áreas de memória.

Outro método utilizado é a subdivisão do programa em várias partes, com a colocação de grandes áreas de trabalho em COMMON. Os módulos assim criados devem ser reordenados pelo programa de ligação, a fim de proporcionarem melhores referências a estas áreas. Este método só pode ser aplicado em programas Fortran através do uso do compilador otimizador especial, "Fortran H Extended[9]", com implicações de custo adicional.*

É evidente, a exemplo do problema de otimização, apresentada em III.3, que esta também não é a melhor solução. Mais uma vez, o programador não recebe uma resposta clara e precisa de seu esforço adicional.

A discussão pormenorizada da eficiência da programação Fortran, em sistemas virtuais, foge ao limite deste trabalho. HATFIELD e GERALD[10] descrevem as técnicas de reestruturação de programas para execução em memória virtual,

---

* Em fevereiro de 1976, o preço do aluguel mensal cobrado pela IBM para o compilador Fortran H Extended e das respectivas livrarias de subprogramas era de Cr$ 3.970,00.

enquanto SCHEDLER e YANG[11] se preocupam com a simulação de um modelo para eficiência em sistemas paginados. A programação estruturada para sistemas virtuais é discutida por ROGERS[12] para as mais importantes linguagens de programação de alto nível.

VI.4 - OUTRAS OTIMIZAÇÕES

Descrevemos aqui as otimizações que se pode fazer, tendo em vista o código-objeto gerado pelo compilador Fortran G.

Os comandos de IF compostos requerem mais memória e maior tempo de execução. Mesmo em prejuízo da elegância, devemos substituí-los por vários comandos de IF lógicos, não compostos.

Os comandos de IF lógicos são mais rápidos que os IF aritméticos. O comando de GO TO calculado tem um tempo fixo e só deve ser utilizado quando houver um grande número de opções de desvio.

O uso de comando DO para o controle de iterações é sensivelmente melhor que aqueles apoiados em testes aritméticos. O compilador G, em uma de suas fases, procura otimizar os comandos dentro do laço e proceder a uma melhor alocação de registradores. O controle do laço tem um tempo fixo. Desta maneira deve-se procurar fazer o maior número de operações dentro de um único comando. Isto também não é o

procedimento mais elegante, mas produz mais eficiência.

O compilador G permite quase todos os tipos de expressão aritmética como subscrito. No entanto, um código-objeto mais eficiente é gerado quando se codifica(variável ± constante), em vez de (constante ± variável), embora ambas as expressões sejam algebricamente equivalentes. É oportuno lembrar que o Fortran armazena os conjuntos multidimensionais de maneira linear, coluna por coluna. Deste modo, a referência a determinado subscrito deve ser calculada para transformá-lo em um endereço de memória. Os compiladores, ao encontrar estas referências, geram o código necessário para o respectivo cálculo, que é feito em tempo de execução. Podemos, então, afirmar em benefício de eficiência, que o uso de conjuntos multidimensionais deve ser restrito à absoluta necessidade destes, nos algorítmos usados.

VII. CONCLUSÕES

Este trabalho não pretende esgotar os problemas relacionados com a eficiência das programações. De fato, dependendo do tipo de aplicação, outras técnicas de otimização podem ser mais eficazes.

O aperfeiçoamento dos sitemas operacionais e da arquitetura dos computadores não foi devidamente acompanhado pelo aperfeiçoamento dos compiladores. Ao contrário, o programador preocupado em eficiência tem hoje, com os novos sistemas, mais problemas e trabalho.

De qualquer modo, acreditamos que nosso trabalho proporcionará ao programador Fortran elementos para melhor eficiência de suas programações, enquanto não se dispuser de compiladores mais compatíveis com a atual tecnologia.

Em vista dos resultados obtidos, neste trabalho, sugerimos para posterior estudo, a aplicação da mesma idéia nas linguagens COBOL e PL/1,notadamente para programas a serem executados sobre o esquema virtual.

APÊNDICE 1

PROGRAMA ANATEMP

A listagem do programa ANATEMP a seguir, foi obtida através do procedimento catalogado FORTTC, ou seja , o ANATEMP faz a análise sobre si próprio.

DESCRICAO DOS RESULTADOS OBTIDOS PELO PROGRAMA ANATEMP

1234 5 6 7 8 9

1 A PRESENCA DE UM -B- NESTE CAMPO INDICA QUE UM -BRANCH- PODERA SER TOMADO

2 UM -E- NESTE CAMPO INDICA -REFERENCIA EXTERNA - AO PROGRAMA PRINCIPAL

3 UM -L- INDICA QUE O COMANDO FINALIZA UM -LOOP- OU CONTEM UM -LOOP

4 FORNECE O NUMERO DE -REFERENCIAS EXTERNAS

5 CONTADOR DE LOCALIZACAO RELATIVO AO ZERO AONDE O COMANDO TEM INICIO

6 EM COMANDOS AONDE PODE HAVER DIVERSOS TEMPOS ESTE CAMPO DEFINE O MENOR TEMPO POSSIVEL

7 TEMPO DE EXECUCAO DO COMANDO

8 TEMPO PARA -UMA- ITERACAO DO -LOOP-EXCETO A PRIMEIRA

9 TEMPO A SER ADICIONADO A -8- PARA A DETERMINACAO DO TEMPO DA PRIMEIRA ITERACAO

LINK= TEMPO DE -LINKAGE-

UNIDADE DE TEMPO = MICRO-SEGUNDOS

```
FORTRAN IV G LEVEL  21                MAIN               DATE = 76237          10/58/11            PAGE 0001
   1234                                                                                          5       6       7       8       9
                                                                                              LINK=   24.92

          C                 PROGRAMA DE ANALISE DOS TEMPOS(ANATEMP)
          C
          C
          C
          C  AUTOR -SIMPLICIO LOPES DE FREITAS           RIO, JANEIRO 1976
          C
          C
          C  RESUMO-O CODIGO OBJETO DE PROGRAMAS COMPILADOS PELO FORTRAN IV LEVEL
          C          21 DA IBM COM AS OPCOES SOURCE E LIST, E' ANALISADO ATRAVES
          C          DO CALCULO DOS TEMPOS DAS INSTRUCOES E RELEVANTES INFORMACOES
          C          SAO FORNECIDAS AO PROGRAMADOR.
          C
          C
          C  SUBROTINAS EXTERNAS UTILIZADAS NAO PROVIDAS PELA LIVRARIA FORTRAN
          C          MOVEC01  - MOVE *N* BYTES DE UMA LOCACAO DE MEMORIA PARA OUTRA
          C          EBCDIC01 - FAZ CONVERSOES DE EBCDIC PARA NUMERO E VICE-VERSA.
          C
          C           NOTA     - ESTAS SUBROTINAS ESTAO CODIFICADAS NA LINGUAGEM
          C                      ASSEMBLER PARA IBM 360-370
          C
          C
          C
          C  ARQUIVOS NECESSARIOS -
          C                 FT07-ARQUIVO EM DISCO -200 TRILHAS SAO ALOCADAS(2314)
          C                 FT08-ARQUIVO EM DISCO -200 TRILHAS SAO ALOCADAS(2314)
          C
          C MEMORIA NECESSARIA - 180K BYTES DEVEM SER ALOCADOS.
          C                      O PROGRAMA PODERA' UTILIZAR-SE DE MENOS MEMORIA
          C                      QUANDO NAO HOUVER NECESSIDADE DE SER UTILIZADO
          C                      O DISCO AUXILIAR PARA CONTER INFORMACOES DA
          C                      PARTE 'SOURCE'.
          C
          C
          C
          C
          C
0001            INTEGER OPERD(6),OPERA(6),LOCA(2)/2*'    '/,IDOLA(2),
               1 DELTA(8)/55,28,14,7,3,2,1,0/,DLAB(26)/26*0/,NOVEZ(110)/110*0/,
               2 ISOUR(33)/33*'    '/,ILST(30)/30*'    '/,IPRT(33)/33*'    '/,
               3 IPR(30),NDG(4)/10000,1000,100,10/,NBRR(4)/10,100,1000,10000/
0002            REAL*8 DUMMY(15)
0003            INTEGER MN/1/,IDIS/0/,KM/34/,NBUF/1/,ISTK/0/,IMO/0/,JMN/0/,I3/3/,
               1 NDOS/0/,NI/-2/,IBXLE/0/,IBALR/0/,KCH/0/,ICNT/0/,IBCR/0/,IFAG/0/,
               2 J/0/,I/0/,NVA/0/,IT/0/,NBR/0/,M/0/,IDL/0/,NCIG/0/,LJ/0/,
               3 ION/'ION '/,LOOP/'BLE '/,ASTEIX/'****'/,ORTR/'ORTR'/,IBLK/'    '/
               4 ,PAREN/'()  '/
0004            REAL TTO/0.0/,TMIN/0.0/,FPSTK(26)/26*0.0/,DOSTA(26)/26*0.0/,
               *  FPZ/0.0/,TOTALG/0.0/
0005            EQUIVALENCE (IPRT(1),IPR(1),DUMMY(1))
          C
          C-------------------------------------------TABELA TEMPOS DAS INSTRUCOES
          C
0006            REAL TEMPO(110)/
               1   1.40,  2.45,  1.72,  2.43,  1.68,  1.80,  1.40,  0.65,  0.65,
               2   2.38,  1.64,  2.40,  1.65,  1.20,  1.20,  0.80,  0.70,  1.15,
               3   0.98,  1.40,  1.40,  1.40,  2.00,  1.26,  1.98,  1.24,  1.80,
               4   1.40,  1.40,  0.65,  0.65,  8.70,  0.00, 14.10, 13.35,  7.30,
               5   6.55,  8.40,  0.00,  1.25,  1.05,  1.40,  1.20,  0.75,  1.05,
```

```
FORTRAN IV G LEVEL  21                  MAIN                DATE = 76237          10/58/11                PAGE 0002

                6   0.85,  0.65,  1.40,  1.05,  1.20,  0.65,  1.40,  0.00,  1.05,  00005800
                7   0.85,  0.95,  1.05,  0.85,  0.95,  2.20,  0.65,  1.05,  0.85,  00005900
                8   0.65,  4.80,  7.60,  7.25,  4.40,  4.08,  5.00,  4.45,  3.70,  00006000
                9   1.33,  2.00,  1.73,  1.25,  2.00,  1.73,  1.25,  1.40,  2.45,  00006100
                A   1.72,  2.43,  1.68,  1.80,  1.40,  0.70,  0.90,  0.90,  0.70,  00006200
                B   0.65,  0.65,  0.90,  0.90,  0.90,  0.90,  0.93,  1.33,  0.93,  00006300
                C   0.93,  1.73,  0.00,  2.38,  1.64,  2.40,  1.65,  1.60,  2.00,  00006400
                D   1.73,  1.25 /                                                   00006500
          C                                                                         00006600
          C---------------          ==============TABELA DAS INSTRUCOES             00006700
          C                                                                         00006800
0007            INTEGER TAB(110)/                                                   00006900
               1 'A   ','AD  ','ADR ','AE  ','AER ','AH  ','AL  ','ALR ','AR  ',  00007000
               2 'AU  ','AUR ','AW  ','AWR ','BAL ','BALR','BC  ','BCR ','BCT ',  00007100
               3 'BCTR','BXH ','BXLE','C   ','CD  ','CDR ','CE  ','CER ','CH  ',  00007200
               4 'CL  ','CLI ','CLR ','CR  ','D   ','DC  ','DD  ','DDR ','DE  ',  00007300
               5 'DER ','DR  ','END ','HDR ','HER ','IC  ','L   ','LA  ','LCDR',  00007400
               6 'LCER','LCR ','LD  ','LDR ','LE  ','LER ','LH  ','LM  ','LNDR',  00007500
               7 'LNER','LNR ','LPDR','LPER','LPR ','LPSW','LR  ','LTDR','LTER',  00007600
               8 'LTR ','M   ','MD  ','MDR ','ME  ','MER ','MH  ','MR  ','MVC ',  00007700
               9 'MVI ','N   ','NI  ','NR  ','D   ','OI  ','OR  ','S   ','SO  ',  00007800
               A 'SDR ','SE  ','SER ','SH  ','SL  ','SLA ','SLDA','SLDL','SLL ',  00007900
               B 'SLR ','SR  ','SRA ','SRDA','SRDL','SRL ','ST  ','STC ','STD ',  00008000
               C 'STE ','STH ','STM ','SU  ','SUR ','SW  ','SWR ','TM  ','X   ',  00008100
               D 'XI  ','XR  ' /                                                   00008200
          C                                                                         00008300
          C                                                                         00008400
          C    AREA E' O VETOR QUE CONTERA' A PARTE 'SOURCE'. ESTA' ASSOCIADA       00008500
          C    A VARIAVEL 'TAM'. SE DESEJAR-SE DIMINUIR A REGIAO ONDE ESTE          00008600
          C    PROGRAMA EXECUTA AMBOS OS VALORES DEVEM SER DIMINUIDOS.              00008700
          C     AREA =(TAM+1)*33 = 26433                                            00008800
          C                                                                         00008900
0008            INTEGER TAM/800/,AREA(26433)/26433*'                                00009000
0009  E 2       WRITE(3,10)                                                         00009100  018468          3.60
0010  E 2       WRITE(3,1190)                                                       00009200  01847C          3.60
0011  E 2       WRITE(3,20)                                                         00009300  018490          3.60
0012          10 FORMAT('1',////////////////////,                                   00009400
              * 25X,'DESCRICAO DOS RESULTADOS OBTIDOS PELO PROGRAMA ANATEMP',///)  00009500
0013          20 FORMAT('-',24X, '1  A PRESENCA DE UM -8- NESTE CAMPO INDICA QUE UM00009600
              1 -BRANCH- PODERA SER TOMADO ',//,                                    00009700
              2 25X,'2  UM -E- NESTE CAMPO INDICA -REFERENCIA EXTERNA - AO PROGRA00009800
              3MA PRINCIPAL ',//,                                                   00009900
              4 25X,'3  UM -L- INDICA QUE O COMANDO FINALIZA UM -LOOP- OU CONTEM 00010000
              5UM -LOOP ',//,                                                       00010100
              6 25X,'4  FORNECE O NUMERO DE -REFERENCIAS EXTERNAS ',//,           00010200
              7 25X,'5  CONTADOR DE LOCALIZACAO RELATIVO AO ZERO ONDE O COMANDO 00010300
              8TEM INICIO ',//,                                                     00010400
              9 25X,'6  EM COMANDOS ONDE PODE HAVER DIVERSOS TEMPOS ESTE CAMPO D00010500
              AEFINE O MENOR TEMPO POSSIVEL ',//,                                   00010600
              B 25X,'7  TEMPO DE EXECUCAO DO COMANDO ',//,                          00010700
              C 25X,'8  TEMPO PARA -UMA- ITERACAO DO -LOOP-EXCETO A PRIMEIRA',//,00010800
              D 25X,'9  TEMPO A SER ADICIONADO A -8- PARA A DETERMINACAO DO TEMPO00010900
              E DA PRIMEIRA ITERACAO ',//,                                          00011000
              F 25X,' LINK=  TEMPO DE -LINKAGE-  ',////,                            00011100
              G 50X,''UNIDADE DE TEMPO = MICRO-SEGUNDOS  ')                         00011200
          C                                                                         00011300
          C                                                                         00011400
          C    LE FTC7 ATE ENCONTRAR 'LIST'                                         00011500
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/58/11          PAGE 0003

                C
                C
                C     COMECA LENDO PARA A MEMORIA
                C
0014  E  3        30 READ(7,40,END=1100,ERR=1030)  IPR
0015              40 FORMAT(80A4)
0016 B               IF(IPRT(3).EQ.ION) GO TO 80
0017 B               IF(ICNT.GE.TAM) GO TO 60
                C
                C     ICNT NUMERO LINHAS NA MEMORIA
                C
0018                 ICNT       =ICNT + DELTA(7)
                C
                C     A CHAMADA A SUBROTINA MOVE PODE COM PREJUIZO NA EFICIENCIA
                C     SER SUBSTITUIDA PELA CODIFICACAO FORTRAN A SEGUIR
                C            DO 1011  I=1,80,3
                C            AREA(MN) = IPR(I)
                C            AREA(MN+1) = IPR(I+1)
                C            AREA(MN+2) = IPR(I+2)
                C       1011 MN=MN + 3
                C
0019  E  1           CALL MOVE(IPRT,AREA(MN))
0020                 MN = MN + 33
0021 B               GO TO 30
0022  E  3        50 READ(7,40,END=1100,ERR=1030) IPR
0023 B               IF(IPRT(3).EQ.ION) GO TO 70
0024  E  3        60 WRITE(8) IPRT
0025                 IDIS = IDIS + DELTA(7)
0026 B               GO TO 50
0027  E  1        70 BACKSPACE 8
0028  E  1           END FILE 8
0029  E  1           REWIND 8
                C
                C     ENCONTRADO INICIO DE 'LIST'
                C     CONTINUA LENDO DE FT07 AGORA TRABALHANDO COM 'LIST'
                C
0030  E  3        80 READ(7,40)  IPR
0031 B               IF( IPRT(2).EQ.ORTR)  GO TO 30
0032  E  1           CALL MOVEC( IPRT(3),3,MNEN,1,4)
                C
                C     ALGORITMO PESQUISA DO 'OPCODE' -KNUTH VOL.III PAG.412
                C
                C     C1-INICIALIZACAO
0033                 I    =  DELTA(1)
0034                 J    =  DELTA(6)
                C     C2-COMPARACAO
0035 B           90 IF( MNEN - TAB(I)) 100,140,110
                C     C3-DECREMENTA 'I'
0036 B          100 IF(DELTA(J).EQ.DELTA(8)) GO TO 120
0037                 I  =  I -DELTA(J)
0038                 J  =  J + DELTA(7)
0039 B               GO TO  90
                C     C4-AUMENTA 'I'
0040 B          110 IF(DELTA(J).EQ.DELTA(8)) GO TO 120
0041                 I  =  I + DELTA(J)
0042                 J  =  J +DELTA(7)
0043 B               GO TO 90
                C
                C     PARA O CASO DA INSTRUCAO NAO TER SIDO ENCONTRADA

00011600
00011700
00011800
00011900  01B4A4                4.80
00012000
00012100  01B4CC                4.80
00012200  01B4DA                4.80
00012300
00012400
00012500
00012600  01B4EB                3.53
00012700
00012800
00012900
00013000
00013100
00013200
00013300
00013400
00013500
00013600  01B4F4               12.01
00013700  01B51E                3.53
00013800  01B52A                2.20
00013900  01B530                4.80
00014000  01B558                4.80
00014100  01B566                5.50
00014200  01B584                3.53
00014300  01B590                2.20
00014400  01B596                3.10
00014500  01B5A4                2.40
00014600  01B5B0                2.40
00014700
00014800
00014900
00015000
00015100  01B58C                4.80
00015200  01B5DC                4.80
00015300  01B5EA                4.83
00015400
00015500
00015600
00015700
00015800  01B5FC                2.13
00015900  01B604                2.13
00016000
00016100  01B60C   7.50         9.70
00016200
00016300  01B62A                6.85
00016400  01B640                3.68
00016500  01B64C                3.53
00016600  01B658                2.20
00016700
00016800  01B65E                6.85
00016900  01B674                3.68
00017000  01B680                3.53
00017100  01B68C                2.20
00017200
00017300
```

```
FJRTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/56/11            PAGE 0004

                C                                                                                  00017400
0044  E  3        120 WRITE(3,130) MNEN                                                            00017500  018692            5.50
0045              130 FORMAT('-',5X,'ERRO/INSTRUCAO  ',A4,' NAO DEFINIDA ')                        00017600
0046  B               GO TO 80                                                                     00017700  018680            2.20
                C                                                                                  00017800
                C     SOMA AO CONTADOR DE INSTRUCOES E DESVIA PARA PROCESSAMENTO DE               00017900
                C     ACORDO COM TIPO DE INSTRUCAO                                               00018000
                C                                                                                  00018100
0047              140 NOVEZ(I) = NOVEZ(I) + DELTA(7)                                             00018200  018686            5.73
0048  B               GO TO (                                                                      00018300  018CA            11.60
                   1    150,   150,   170,   150,   17),   150,   150,   170,   170,             00018400
                   2    150,   170,   150,   170,   53),   510,   550,   460,   590,             00018500
                   3    610,   170,   540,   150,   15),   170,   150,   170,   150,             00018600
                   4    150,   170,   170,   170,   15),    80,   150,   170,   150,             00018700
                   5    170,   170,   200,   170,   17),   150,   150,   150,   170,             00018800
                   6    170,   170,   150,   170,   15),   170,   150,   900,   170,             00018900
                   7    170,   170,   170,   170,   17),   170,   170,   170,   170,             00019000
                   8    170,   150,   150,   170,   150,   170,   150,   170,   170,             00019100
                   9    170,   150,   170,   170,   15),   170,   170,   150,   150,             00019200
                   A    170,   150,   170,   150,   15),   630,   680,   680,   630,             00019300
                   B    170,   170,   740,   790,   79),   740,   150,   150,   150,             00019400
                   C    150,   150,   910,   150,   17),   150,   170,   170,   150,             00019500
                   D    170,   170 ),I                                                           00019600
                C                                                                                  00019700
                C     INSTRUCOES QUE PODEM SER INDEXADAS                                         00019800
                C                                                                                  00019900
0049              150 J  = DELTA(6)                                                              00020000  0188A8            2.13
0050  E  1        160 CALL MOVEC(IPRT(12),J,KCH,4,1)                                             00020100  018880            4.83
0051                  J  =  J + DELTA(7)                                                           00020200  0188C2            3.53
0052  B               IF( KCH.NE.77) GO TO 160                                                     00020300  0188CE            6.00
0053  E  1            CALL MOVEC(IPRT(12),J,KCH,4,1)                                             00020400  0188E0            4.83
0054  B               IF(KCH.EQ.240) GO TO 170                                                     00020500  0188F2            4.80
0055                  TIME  = TEMPO(I) + 0.15                                                      00020600  018900            6.61
0056  B               GO TO 180                                                                    00020700  018914            2.20
                C                                                                                  00020800
                C     INSTRUCOES CUJO TEMPO E' FIXO                                              00020900
                C                                                                                  00021000
0057              170 TIME  = TEMPO (I)                                                          00021100  01891A            4.18
                C                                                                                  00021200
                C     VERIFICA EM 'LIST' SE HA NUMERO DO STAT.NA INSTRUCAO CORRENTE               00021300
                C                                                                                  00021400
0058  E  1        180 CALL MOVEC(IPRT(4),4,ISTANI,1,4)                                           00021500  01892A            4.83
0059  E  1            CALL DEBC (ISTANI,'I',ISTA,4)                                              00021600  01893C            3.15
0060  B               IF( ISTA.NE.DELTA(8)) GO TO 200                                            00021700  018946            4.80
                C                                                                                  00021800
                C     SOMA TEMPO ACUMULADO PARA STATMENT                                         00021900
                C                                                                                  00022000
0061              190 TTO = TTO + TIME                                                           00022100  018954            4.56
0062                  TOTALG = TOTALG  + TIME                                                      00022200  018960            4.56
                C                                                                                  00022300
                C     IFAG=1 INTERROMPE A SOMA DO ACUMULADO PARA TEMPO MINIMO                     00022400
                C                                                                                  00022500
0063  B               IF(IFAG.EQ.DELTA(7)) GO TO 80                                              00022600  01896C            4.80
0064                  TMIN = TMIN + TIME                                                           00022700  01897A            4.56
                C                                                                                  00022800
                C     VERIFICA SE VAI INTERROMPER O ACUMULADO PARA TEMPO MINIMO                   00022900
                C                                                                                  00023000
0065  B               IF(I8CR.EQ.DELTA(7)) IFAG= DELTA(7)                                        00023100  018986    4.80    6.93
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/58/11.          PAGE 0005

0066 B                GO TO 80                                                00023200  01B99C            2.20
                C                                                             00023300
                C     SE IT=0 O TEMPO DE 'LINK' VAI SER CALCULADO             00023400
                C                                                             00023500
0067 B            200 IF( IT.EQ.DELTA(3)) GO TO 250                           00023600  01B9A2            4.80
                C                                                             00023700
                C     TRANSFERE 'LOCATION' PARA LISTAGEM                      00023800
                C                                                             00023900
0068   E  1           CALL MOVEC(LOCA,3,ISOUR(25),3,6)                       00024000  01B9B0            4.83
                C                                                             00024100
                C     TRANSFERE TEMPO ACUMULADO PARA LISTAGEM                 00024200
                C                                                             00024300
0069   E  1           CALL PAREBC( TTO,'F',ISOUR(29),7,2)                    00024400  01B9C2            4.83
                C                                                             00024500
                C     VERIFICA SE HOUVE CHAMADAS EXTERNAS                     00024600
                C                                                             00024700
0070 B                IF(IBALE.EQ.DELTA(8)) GO TO 210                         00024800  01B9D4            4.80
0071   E  1           CALL PAREBC(IBALE,'I',ISOUR(4),2)                      00024900  01B9E2            4.83
0072   E  1           CALL MOVEC (LOOP,3,ISOUR(3),3,1)                       00025000  01B9F4            4.83
                C                                                             00025100
                C     VERIFICA SE HA CONDICAO DE 'LOOP'                       00025200
                C                                                             00025300
0073 B            210 IF(IBXLE.EQ.DELTA(8)) GO TO 220                         00025400  01BA06            4.80
0074   E  1           CALL MOVEC (LOOP,2,ISOUR(3),4,1)                       00025500  01BA14            4.83
                C                                                             00025600
                C     VERIFICA SE TEMPO MINIMO DEVE SER COLOCADO              00025700
                C                                                             00025800
0075 B            220 IF( TTO.EQ.TMIN) GO TO 230                              00025900  01BA26            5.38
0076 B                IF( TMIN.EQ.0.0) GO TO 230                              00026000  01BA34            6.58
0077   E  1           CALL PAREBC(TMIN,'F',ISOUR(27),7,2)                    00026100  01BA46            4.83
                C                                                             00026200
                C     VERIFICA SE HA SITUACAO DE 'BRANCH'                     00026300
                C                                                             00026400
0078 BE 1         230 IF(IBCR.NE.DELTA(3)) CALL MOVEC(LOOP,1,ISOUR(3),2,1)  00026500  01BA58   4.80     9.63
0079                  IFAG = DELTA(8)                                         00026600  01BA78            2.13
                C                                                             00026700
                C     VERIFICA SE O TEMPO DESTE STAT. VAI PARA O 'STACK'      00026800
                C                                                             00026900
0080 B                IF(NODS.EQ.DELTA(7)) DOSTA(ISTK) = TTO                  00027000  01BA80   4.80     8.98
0081                  NODS = DELTA(8)                                         00027100  01BA9E            2.13
                C                                                             00027200
                C     VERIFICA SE HA ELEMENTOS NO STACK                       00027300
                C                                                             00027400
0082 B                IF( ISTK.EQ.DELTA(8)) GO TO 270                         00027500  01BAA6            4.80
                C                                                             00027600
                C     SOMA TEMPO A TODOS ELEMENTOS QUE ESTAO NO STACK         00027700
                C                                                             00027800
0083                  DO 240  LJ=1,ISTK                                       00027900  01BAB4            4.53
0084   L          240 FPSTK(LJ)  = FPSTK(LJ) + TTO                           00028000  01BAC4           12.91(  13.84) 3.6
                C                                                             00028100
                C     VERIFICA SE O LABEL ENCONTRADO ESTA CONTIDO NO STACK     00028200
                C                                                             00028300
0085 B                IF( IMO.NE.DELTA(7)) GO TO 270                          00028400  01BAEC            4.80
0086                  TTOI = FPSTK(ISTK) - DOSTA(ISTK) + TEMPO(99)            00028500  01BAFA            9.19
0087                  DSTA = DOSTA(ISTK) - TEMPO(99)                          00028600  01BB12            4.71
                C                                                             00028700
                C     COLOCA TEMPOS 'DO' PARA LISTAGEM                        00028800
                C                                                             00028900
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/58/11              PAGE 0006

0088  E  1        CALL PAREBC( TTOI,'F',ISOUR(31),7,2)                          00029000  01BB1E          4.88
0089  E  1        CALL PAREBC( DSTA,'F',ISOUR(33),4,1)                          00029100  01BB30          4.88
                C                                                               00029200
                C     COLOCA PARENTESES  NO TEMPO 'DO'                          00029300
                C                                                               00029400
0090  E  1        CALL MOVEC(PAREN,1,ISOUR(30),4,1)                             00029500  01BB42          4.88
0091  E  1        CALL MOVEC(PAREN,2,ISOUR(32),4,1)                             00029600  01BB54          4.88
0092              IDL  =  DLAB(ISTK)                                            00029700  01BB66          2.28
0093              DLAB(ISTK) = DELTA(8)                                         00029800  01BB6E          2.28
0094              FPSTK(ISTK)= FPZ                                              00029900  01BB76          2.28
0095              DSTA(ISTK)= FPZ                                               00030000  01BB7E          2.28
0096              IMO  =  DELTA(8)                                              00030100  01BB86          2.13
0097              ISTK =  ISTK - DELTA(7)                                       00030200  01BB8E          3.53
0098  B         .IF( JMN.NE.DELTA(7)) GO TO 270                                 00030300  01BB9A          4.80
0099  E  5        WRITE(3,260) ISOUR(31),ISOUR(32),ISOUR(33)                    00030400  01BBA8          7.20
0100          260 FORMAT(119X,'(',3A4)                                          00030500
0101              JMN = DELTA(8)                                                00030600  01BBD4          2.13
0102  B           GO TO 280                                                     00030700  01BBDC          2.20
0103  E  3    270 WRITE(3,310) ISOUR                                            00030800  01BBE2          5.50
0104  B       280 IF(ISTK.EQ.DELTA(3)) GO TO 320                                00030900  01BC04          4.80
0105  B           IF(DLAB(ISTK).NE.IDL)GO TO 320                                00031000  01BC12          6.85
                C                                                               00031100
                C     SETA CHAVE INDICANDO QUE O LABEL ENCONTRADO FECHA OUTRO 'DO'  00031200
                C                                                               00031300
0106              JMN= DELTA(7)                                                 00031400  01BC28          2.13
0107  B           GO TO 250                                                     00031500  01BC30          2.20
0108          290 IT = DELTA(7)                                                 00031600  01BC36          2.13
                C                                                               00031700
0109  EL 3        WRITE(3,310) (AREA(I),I=1,33)                                 00031800  01BC3E         19.83
0110  BE  2       IF(I3.EQ.DELTA(3)) WRITE(3,1190)                              00031900  01BC90    4.80  9.10
0111  E  3        WRITE(3,300) TTO                                              00032000  01BCB4          4.80
0112          300 FORMAT( 98X,'LINK= ',F7.2)                                    00032100
0113          310 FORMAT(3344)                                                  00032200
0114          320 TTO = FPZ                                                     00032300  01BCD0          2.13
0115              TMIN = FPZ                                                    00032400  01BCD8          2.13
0116              I8CR = DELTA(8)                                               00032500  01BCE0          2.13
0117              IBXLE= DELTA(8)                                               00032600  01BCE8          2.13
0118              IBALR = DELTA(8)                                              00032700  01BCF0          2.13
0119  B           GO TO 340                                                     00032800  01BCF8          2.20
                C                                                               00032900
0120  E  3    330 WRITE(3,310) ISOUR                                            00033000  01BCFE          5.50
                C                                                               00033100
                C     VERIFICA SE TRABALHA COM DADOS NA MEMORIA OU DISCO         00033200
                C                                                               00033300
0121  B       340 IF(NBUF.LE.TAM) GO TO 350                                     00033400  01BD20          4.80
                C                                                               00033500
                C     TRABALHANDO COM DADOS NO DISCO                            00033600
                C                                                               00033700
0122  E  3        READ(8) ISOUR                                                 00033800  01BD2E          5.50
0123  B           GO TO 360                                                     00033900  01BD4C          2.20
                C                                                               00034000
                C     TRABALHANDO COM DADOS NA MEMORIA                          00034100
                C                                                               00034200
0124  E  1    350 CALL MOVE(AREA(KM),ISOUR)                                     00034300  01BD52         10.68
0125              KM = KM + 33                                                  00034400  01BD78          3.53
                C                                                               00034500
                C     NBUF=NUMERO DE LINHAS NA MEMORIA JA TRABALHADAS           00034600
                C                                                               00034700
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/58/11             PAGE 0007

0126                    NBUF = NBUF + DELTA(7)                                00C348C0  01BD84          3.53
                C
                C       ELIMINA STAT. NAO EXECUTAVEIS                         00035000
                C                                                             00035100
0127 B            360 IF(ISOUR(2).EQ.ORTR) GO TO 330                        00035200  01BD90          4.80
0128 B                IF(ISOUR(2).EQ.ASTEIX) GO TO 330                      00035300  01BD9E          4.80
0129  E  1            CALL DEBC(ISOUR(2),'I',IFOR,4)                        00035400  01BDAC          4.83
                C                                                             00035500
                C       VERIFICA SE STAT.DE LIST IGUAL AO DE SOURCE           00C35600
                C                                                             00035700
0130 B                IF(IFOR.NE.ISTA) GO TO 330                            00035800  01BDBE          4.80
                C                                                             00035900
                C       ENCONTRADO STAT.EM SOURCE                             00036000
                C                                                             00036100
                C       MOVE LOCACAO PARA LUGAR AUXILIAR                      00036200
                C                                                             00036300
0131  E  1            CALL MOVEC(IPR (2),1,LOCA,3,5)                        00036400  01BDCC          4.83
0132 B                IF(ISTK.EQ.0) GO TO 390                               00036500  01BDDE          6.00
                C                                                             00036600
                C       O STACK CONTEM 'LABELS'                               00036700
                C                                                             D0036800
0133  E  1            CALL DEBC(ISOUR(5),'I',LABEL,5)                       00036900  01BDF0          4.83
0134 B                IF(LABEL.EQ.0) GO TO 390                              00037000  01BE02          4.80
                C                                                             00037100
                C       DETERMINA O LABEL(COL.17 A 21)                        00037200
                C                                                             00037300
0135                    NBR = DELTA(8)                                        00037400  01BE10          2.13
0136                    DO 370 J=1,4                                          00037500  01BE18          3.33
0137                    M  = 6 - J                                            00037600  01BE24          4.73
0138  E  1              CALL MOVEC(ISOUR(5),M,KCH,4,1)                        00037700  01BE34          4.83
0139 B                  IF(KCH.NE.64) GO TO 380                               00037800  01BE46          4.80
0140   L          370 NBR = NBR + DELTA(7)                                   00037900  01BE54          8.83(  24.12) 2.4
0141 B            380 IF(NBR.GE.1) LABEL= LABEL / NBRR(NBR)                  00038000  01BE74    6.00  22.98
                C                                                             00038100
                C       VERIFICA SE LABEL ESTA NO STACK - SE POSITIVO SETA FLAG00038200
                C                                                             00038300
0142 B                  IF(LABEL.EQ.OLAB(ISTK)) IMO= DELTA(7)                 00038400  01BE9E    6.85  8.98
0143              390 J= DELTA(5)                                             00038500  01BEBC          2.13
0144  E  1        400 CALL MOVEC(ISOUR(6),J,KCH,4,1)                        00038600  01BEC4          4.83
0145                    J= J + DELTA(7)                                       00038700  01BED6          3.53
                C                                                             00038800
                C       VERIFICA SE STATMENT E'  'DO'                         00038900
                C                                                             00039000
                C       COMPARA COM 'ESPACO'(BRANCO)                          00039100
                C                                                             00039200
0146 B                  IF(KCH.EQ.64) GO TO 400                               00039300  01BEE2          6.00
                C                                                             00039400
                C       COMPARA COM 'O'                                       00039500
                C                                                             00039600
0147 B                  IF(KCH.NE.196)GO TO 450                               00039700  01BEF4          4.80
0148                    J= J - DELTA(7)                                       00039800  01BF02          3.53
0149              410 J= J + DELTA(7)                                         00039900  01BF0E          3.53
0150  E  1              CALL MOVEC(ISOUR(5),J,KCH,4,1)                        00040000  01BF1A          4.83
0151 B                  IF(KCH.EQ.64) GO TO 410                               00040100  01BF2C          6.00
                C                                                             00040200
                C       COMPARA COM 'O'                                       00040300
                C                                                             00040400
0152 B                  IF(KCH.NE.214)GO TO 450                               00040500  01BF3E          4.80
```

```
FORTRAN IV G LEVEL  21                MAIN                DATE = 76237        10/58/11              PAGE 0008

0153            420 J= J + DELTA(7)                                          00040600  01BF4C            3.53
0154  E  1          CALL MOVEC(ISOUR(5),J,KCH,4,1)                         00040700  01BF58            4.83
                C                                                          00040800
                C     VERIFICA SE E' ESPACO OU NUMERICO                    00040900
                C                                                          00041000
0155 B              IF(KCH.EQ.64) GO TO 420                                00041100  01BF6A            6.00
0156 B              IF(KCH.LT.240)GO TO 450                                00041200  01BF7C            4.80
0157                IDOLA(1)=  IBLK                                        00041300  01BF8A            2.13
0158                IDOLA(2)=  IBLK                                        00041400  01BF92            2.13
0159                NDIG    =  DELTA(8)                                    00041500  01BF9A            2.13
0160                IOTO    =  DELTA(5)                                    00041600  01BFA2            2.13
0161            430 IOTO    =  IOTO  + DELTA(7)                            00041700  01BFAA            3.53
0162                NDIG    =  NDIG  + DELTA(7)                            00041800  01BFB6            3.53
0163  E  1          CALL MOVEC(ISOUR(6),J,IDOLA,IOTO,1)                    00041900  01BFC2            4.83
0164                J       =  J + DELTA(7)                                00042000  01BFD4            3.53
0165  E  1          CALL MOVEC(ISOUR(5),J,KCH,4,1)                         00042100  01BFE0            4.83
0166 B              IF(KCH.GE.240)GO TO 430                                00042200  01BFF2            6.00
                C                                                          00042300
                C     CONVERTE NUMERO DO LABEL CONTIDO NO STAT.'DO'        00042400
                C                                                          00042500
0167  E  1          CALL DEBC(IDOLA,'I',IDOLB,8)                           00042600  01CCC4            3.15
0168 B              IF(NDIG.LT.5) IDOLB = IDOLB / NDG(NDIG)                00042700  01CC0E    4.80   21.78
0169            440 J  = J + DELTA(7)                                      00042800  01C034            3.53
                C                                                          00042900
                C     PESQUISA SO ATE COLUNA 72                            00043000
                C                                                          00043100
0170 B              IF(J.GT.65) GO TO 450                                  00043200  01CC40            6.00
0171  E  1          CALL MOVEC(ISOUR(5),J,KCH,4,1)                         00043300  01CC52            4.83
                C                                                          00043400
                C     COMPARA COM 'ABRE PARENTESES'                        00043500
                C                                                          00043600
0172 B              IF(KCH.EQ.77) GO TO 450                                00043700  01CC64            4.80
                C                                                          00043800
                C     COMPARA COM VIRGULA                                  00043900
                C                                                          00044000
0173 B              IF(KCH.NE.107)GO TO 440                                00044100  01CC72            4.80
                C                                                          00044200
                C     AUMENTE O STACK                                      00044300
                C                                                          00044400
0174                ISTK = ISTK + DELTA(7)                                 00044500  01CC80            3.53
                C                                                          00044600
                C     COLOCA LABEL DO STAT.'DO' NO STACK                   00044700
                C                                                          00044800
0175                DLAB(ISTK) = IDOLB                                     00044900  01CC8C            4.18
                C                                                          00045000
                C     ACIONA FLAG                                          00045100
                C                                                          00045200
0176                NDOS = DELTA(7)                                        00045300  01CC9C            2.13
                C                                                          00045400
                C     VERIFICA SE 'END' FOI ENCONTRADO                     00045500
                C                                                          00045600
0177 B          450 IF(I.NE.39) GO TO 190                                  00045700  01C0A4            6.00
0178 B              GO TO 1110                                             00045800  01C0B6            2.20
                C                                                          00045900
                C     CALCULO DOS TEMPOS DOS OUTROS TIPOS DE INSTRUCOES    00046000
                C                                                          00046100
                C                                                          00046200
                C     BCR                                                  00046300
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237        10/56/11          PAGE 0009

                C                                                                    00046400
0179              460 ASSIGN 470 TO JSUB                                             00046500  01C0BC          2.13
0180 B              GO TO 1010                                                       00046600  01C0C4          2.20
0181 B            470 IF(OPERA(1).GT.DELTA(3)) GO TO 490                           00046700  01C0CA          4.80
0182              480 F1=FPZ                                                       00046800  01C0D8          2.13
0183 B              GO TO 500                                                      00046900  01C0E0          2.20
0184 B            490 IF(OPERA(2).LE.DELTA(3)) GO TO 480                           00047000  01C0E6          4.80
                C     ACIONA FLAG                                                  00047100
0185                IBCR= DELTA(7)                                                 00047200  01C0F4          2.13
0186                F1= 0.3                                                        00047300  01C0FC          3.33
0187              500 TIME = TEMPO(I) + F1                                         00047400  01C108          6.61
0188 B              GO TO 180                                                      00047500  01C11C          2.20
                C                                                                  00047600
                C     BALR                                                         00047700
                C                                                                  00047800
0189              510 ASSIGN 520 TO JSUB                                           00047900  01C122          2.13
0190 B              GO TO 1010                                                     00048000  01C12A          2.20
0191              520 TIME=TEMPO(I)                                                00048100  01C130          4.18
                C     SOMA ACUMUL DE REF.EXTERNA SE NECESSARIO                     00048200
0192 B              IF(OPERA(2).NE.DELTA(8)) IBALR = IBALR + DELTA(7)              00048300  01C140    4.80  8.33
0193 B              GO TO 180                                                      00048400  01C15A          2.20
                C                                                                  00048500
                C     BAL                                                          00048600
                C                                                                  00048700
                C     ADICIONA NUMERO DE REF.EXTERNAS                              00048800
0194              530 IBALR= IBALR + DELTA(7)                                      00048900  01C160          3.53
0195 B              GO TO 170                                                      00049000  01C16C          2.20
                C                                                                  00049100
                C     BXLE                                                         00049200
                C                                                                  00049300
                C     ACIONA FLAG INDICANDO LOOP                                   00049400
0196              540 IBXLE = DELTA(7)                                             00049500  01C172          2.13
0197 B              GO TO 170                                                      00049600  01C17A          2.20
                C                                                                  00049700
                C     BC                                                           00049800
                C                                                                  00049900
0198              550 ASSIGN 560 TO JSUB                                           00050000  01C180          2.13
0199 B              GO TO 1010                                                     00050100  01C188          2.20
0200              560 F1= 0.3                                                      00050200  01C18E          3.33
0201 B              IF(OPERA(1).EQ.DELTA(3)) GO TO 570                           00050300  01C19A          4.80
0202 B              IF(OPERA(2).NE.DELTA(3)) GO TO 580                           00050400  01C1A8          4.80
0203              570 F1= FPZ                                                      00050500  01C1B6          2.13
0204              580 TIME = TEMPO(I) + F1                                         00050600  01C1BE          6.61
0205 B              GO TO 180                                                      00050700  01C1D2          2.20
                C                                                                  00050800
                C     BCT                                                          00050900
                C                                                                  00051000
0206              590 ASSIGN 600 TO JSUB                                           00051100  01C1D8          2.13
0207 B              GO TO 1010                                                     00051200  01C1E0          2.20
0208 B            600 IF(OPERA(3).EQ.DELTA(8))GO TO 170                            00051300  01C1E6          4.80
0209                TIME= TEMPO(I) + 0.2                                           00051400  01C1F4          7.81
0210 B              GO TO 180                                                      00051500  01C2CC          2.20
                C                                                                  00051600
                C     BCTR                                                         00051700
                C                                                                  00051800
0211              610 ASSIGN 620 TO JSUB                                           00051900  01C212          2.13
0212 B              GO TO 1010                                                     00052000  01C21A          2.20
0213              620 F1= 0.17                                                     00052100  01C220          3.33
```

```
FORTRAN IV G LEVEL  21                MAIN                DATE = 76237      10/58/11            PAGE 0010

0214 B              IF(OPERA(2).EQ.DELTA(8)) F1= FPZ                          00052200  01C22C   4.80     6.93
0215                TIME = TEMPO(I) + F1                                      00052300  01C242            6.61
0216 B              GO TO 180                                                 00052400  01C256            2.20
                C                                                             00052500
                C     SLA   SLL                                               00052600
                C                                                             00052700
0217              630 ASSIGN 640 TO JSUB                                      00052800  01C25C            2.13
0218 B              GO TO 1010                                                00052900  01C264            2.20
0219              640 IQ4= OPERA(2) / 4                                       00053000  01C26A           16.13
0220                IR4 = MOD (OPERA(2),4)                                    00053100  01C27E           23.91
0221                S1  = FPZ                                                 00053200  01C2A0            2.13
0222 B              IF(IR4.NE.DELTA(5)) GO TO 650                             00053300  01C2A8            4.80
0223 B              IF(IR4.NE.DELTA(3)) GO TO 650                             00053400  01C2B6            4.80
0224                S1  = 2.                                                  00053500  01C2C4            2.13
0225 B              GO TO 670                                                 00053600  01C2CC            2.20
0226 B            650 IF(IR4.EQ.DELTA(5)) GO TO 660                           00053700  01C2D2            4.80
0227 B              IF(IR4.NE.DELTA(3)) GO TO 670                             00053800  01C2F0            4.80
0228              660 S1  = 1.                                                00053900  01C2EE            3.33
0229              670 TIME = TEMPO(I) +(0.2 * FLOAT(IQ4)) + (0.2 * S1)        00054000  01C2FA           29.22
0230 B              GO TO 180                                                 00054100  01C338            2.20
                C                                                             00054200
                C     SLDL  SLDA                                              00054300
                C                                                             00054400
0231              680 ASSIGN 690 TO JSUB                                      00054500  01C33E            2.13
0232 B              GO TO 1010                                                00054600  01C346            2.20
0233              690 IQ4= OPERA(2) / 4                                       00054700  01C34C           16.13
0234                IR4  = MOD(OPERA(2),4)                                    00054800  01C360           23.91
                C     FORMULA 'S3'                                            00054900
0235 B              IF(IR4- DELTA(7)) 700,710,720                             00055000  01C382   5.45     7.65
0236              700 S3 = FPZ                                                00055100  01C398            2.13
0237 B              IF(IQ4.EQ.DELTA(3)) S3=1.                                 00055200  01C3A0   4.80     8.13
0238 B              GO TO 730                                                 00055300  01C3BA            2.20
0239              710 S3 = 3.                                                 00055400  01C3C0            3.33
0240 B              GO TO 730                                                 00055500  01C3CC            2.20
0241              720 S3 = 5.                                                 00055600  01C3D2            3.33
0242              730 TIME= TEMPO(I) + (0.4 *FLOAT(IQ4)) + (0.2 * S3)         00055700  01C3DE           29.22
0243 B              GO TO 180                                                 00055800  01C41C            2.20
                C                                                             00055900
                C     SRA   SLL                                               00056000
                C                                                             00056100
0244              740 ASSIGN 750 TO JSUB                                      00056200  01C422            2.13
0245 B              GO TO 1010                                                00056300  01C42A            2.20
0246              750 IQ4= OPERA(2) / 4                                       00056400  01C430           16.13
0247                IR4 = MOD (OPERA(2),4)                                    00056500  01C444           23.91
                C     FORMULA S2                                              00056600
0248                S2 = FPZ                                                  00056700  01C466            2.13
0249 B              IF(IR4.EQ.DELTA(5)) GO TO 760                             00056800  01C46E            4.80
0250 B              IF(IQ4.NE.DELTA(8)) GO TO 770                             00056900  01C47C            4.80
0251              760 S2 = 1.                                                 00057000  01C48A            3.33
0252 B              GO TO 780                                                 00057100  01C496            2.20
0253 B            770 IF(IR4.NE.DELTA(5)) GO TO 780                           00057200  01C49C            4.80
0254 B              IF(IQ4.NE.DELTA(8)) GO TO 780                             00057300  01C4AA            4.80
0255                S2 = 2.                                                   00057400  01C4B8            3.33
0256              780 TIME = TEMPO(I) + (0.2*FLOAT(IQ4)) + (0.2 * S2)         00057500  01C4C4           29.22
0257 B              GO TO 180                                                 00057600  01C502            2.20
                C                                                             00057700
                C     SROA  SRDL .                                            00057800
                C                                                             00057900
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237      10/56/11          PAGE 0011

0253           790 ASSIGN 800 TO JSUB                                    00058000  01C508          2.13
0254 B             GO TO 1010                                            00058100  01C510          2.20
0255           800 IQ4=OPERA(2)/4                                        00058200  01C516         16.13
0256               IR4 = MOD(OPERA(2),4)                                 00058300  01C52A         23.61
               C   FORMULA S4                                           00058400
0257 B             IF(IR4.EQ.DELTA(3)) GO TO 880                         00058500  01C54C          4.80
0258 B             IF(IQ4.EQ.DELTA(3)) GO TO 840                         00058600  01C55A          4.80
0259 B             IF(IR4 - DELTA(5)) 810,820,830                        00058700  01C568   5.45   7.65
0260           810 S4 = 5.                                               00058800  01C57E          3.33
0261 B             GO TO 890                                             00058900  01C58A          2.20
0262           820 S4 = 4.                                               00059000  01C590          3.33
0263 B             GO TO 890                                             00059100  01C59C          2.20
0264           830 S4= 3.                                                00059200  01C5A2          3.33
0265 B             GO TO 890                                             00059300  01C5AE          2.20
0266           840 IF(IR4 - DELTA(6)) 850,850,870                        00059400  01C5B4   5.45   7.65
0267           850 S4= 4.                                                00059500  01C5CA          3.33
0268 B             GO TO 890                                             00059600  01C5D6          2.20
0269           860 S4= 3.                                                00059700  01C5DC          3.33
0270 B             GO TO 890                                             00059800  01C5E8          2.20
0271           870 S4 = 2.                                               00059900  01C5EE          3.33
0272 B             GO TO 890                                             00060000  01C5FA          2.20
0273           880 S4= FPZ                                               00060100  01C600          2.13
0274           890 TIME=TEMPO(I)+(0.4*FLOAT(IQ4)) +(0.2 * S4)            00060200  01C608         29.22
0275 B             GO TO 180                                             00060300  01C646          2.20
               C                                                         00060400
               C   STM   LM                                             00060500
               C                                                         00060600
0276           900 CONTINUE                                              00060700
0277           910 ASSIGN 920 TO JSUB                                    00060800  01C64C          2.13
0278 B             GO TO 1010                                            00060900  01C654          2.20
0279 B         920 IF(OPERA(1).GT.OPERA(2)) GO TO 930                    00061000  01C65A          4.80
               C   DETERMINA NUMERO REGISTADORES                        00061100
0280               IGR = (OPERA(2) - OPERA(1)) + DELTA(7)                00061200  01C668          4.93
0281 B             GO TO 940                                             00061300  01C678          2.20
0282           930 IGR = (OPERA(2) + 17) - OPERA(1)                      00061400  01C67E          6.13
               C   DET. SE NUMERO REGS. E' PAR OU IMPAR                 00061500
0283           940 IP = MOD(IGR,2)                                       00061600  01C692         22.98
0284 B             IF( IP.EQ.DELTA(3)) GO TO 960                         00061700  01C6B0          4.80
               C   FORMULA 'A4'                                         00061800
0285 B             IF(I.EQ.102) GO TO 950                                00061900  01C6BE          4.80
0286               TIME = 1.0 + (0.3 * IGR)                              00062000  01C6CC         18.69
0287 B             GO TO 180                                             00062100  01C6F4          2.20
0288           950 TIME = 0.93 + (0.2* FLOAT(IGR))                       00062200  01C6FA         19.89
0289 B             GO TO 180                                             00062300  01C726          2.20
               C   DETERMINA SE ESTA' EM LIMITE DE PALAVRA DUPLA         00062400
0290           960 IDW = MOD(OPERA(3),8)                                 00062500  01C72C         25.11
0291 B             IF(IDW.NE.DELTA(3)) GO TO 990                         00062600  01C752          4.80
0292 B             IF(IGR.GT.DELTA(6)) GO TO 970                         00062700  01C760          4.80
0293               TIME = 1.40                                           00062800  01C76E          2.33
0294 B             IF(I.EQ.102) TIME= 1.33                               00062900  01C776   4.80   6.93
0295 B             GO TO 180                                             00063000  01C78C          2.20
0296 B         970 IF(I.EQ.102) GO TO 980                                00063100  01C792          6.00
0297               TIME = 0.8 + (0.3 * IGR)                              00063200  01C7A4         18.69
0298 B             GO TO 180                                             00063300  01C7CC          2.20
0299           980 TIME = (.55 + (0.2 * FLOAT(IGR))                      00063400  01C7D2         19.89
0300 B             GO TO 180                                             00063500  01C7FE          2.20
0301 B         990 IF(I.EQ.102) GO TO 1000                               00063600  01C8C4          6.00
0302               TIME = 1.2 + (0.3 * IGR)                              00063700  01C816         18.69
```

```
FORTRAN IV G LEVEL  21                  MAIN              DATE = 76237        10/58/11          PAGE 0012

0308 B              GO TO 180                                                    00063800  01C83E          2.20
0309           1000 TIME = 1.33 +(0.2 * FLOAT(IGR))                              00063900  01C844         19.89
0310 B              GO TO 180                                                    00064000  01C870          2.20
                C                                                                00064100
                C                                                                00064200
                C     SUBROTINA INTERNA NO PROGRAMA                              00064300
                C                                                                00064400
                C  RESUMO   -DADA QUALQUER INSTRUCAO NO FORMATO DE ASSEMBLER      00064500
                C            FORNECIDA PELA OPCAO DE 'LIST' DO COMPILADOR FORTRAN 'G'  00064600
                C            ESTA SUBROTINA FORNECE NO ARRAY DE SAIDA OS VALORES  00064700
                C            BINARIOS DOS OPERANDOS ENCONTRADOS                  00064800
                C                                                                00064900
                C                OPERJ -ARRAY DE ENTRADA QUE CONTEM INSTRUCAO NO  00065000
                C                       FORMATO ASSEMBLER                        00065100
                C                OPERA -ARRAY DE SAIDA QUE CONTEM O VALOR BINARIO 00065200
                C                       DOS OPERANDOS                            00065300
                C                                                                00065400
                C                                                                00065500
0311  E  1       1010 CALL MOVEC(IPRT(12),2,OPERJ,1,24)                          00065600  01C876          4.83
0312                 ISET = DELTA(8)                                             00065700  01C888          2.13
0313                 JARI = DELTA(7)                                             00065800  01C890          2.13
0314                 DO 1060  N=1,6                                              00065900  01C898          5.73
0315                 OPERA(N) = IBLK                                             00066000  01C8AC          2.13
0316                 NUMBA= DELTA(8)                                             00066100  01C8B4          2.13
0317  E  1       1020 CALL MOVEC(OPERJ,JARI,KC14,4,1)                            00066200  01C88C          3.15
0318                 JARI = JARI + DELTA(7)                                      00066300  01C8C6          3.53
0319 B               IF(KCH.GE.240) GO TO 1040                                   00066400  01C8D2          6.00
0320 B               IF(KCH.EQ.107) GO TO 1050                                   00066500  01C8E4          4.80
0321 B               IF(KCH.EQ. 77) GO TO 1050                                   00066600  01C8F2          4.80
0322 B               IF(KCH.EQ. 93) GO TO 1050                                   00066700  01C900          4.80
0323 B               IF(KCH.EQ. 64) GO TO 1030                                   00066800  01C90E          4.80
0324 B               GO TO 1070                                                  00066900  01C91C          2.20
0325           1030  ISET = DELTA(7)                                             00067000  01C922          2.13
0326 B               GO TO 1050                                                  00067100  01C92A          2.20
0327           1040  NUMBA= NUMBA + DELTA(7)                                     00067200  01C930          3.53
0328 B               GO TO 1020                                                  00067300  01C93C          2.20
0329           1050  INC  = JARI - NUMBA - DELTA(7)                              00067400  01C942          4.93
0330                 IDET = 5  - NUMBA                                           00067500  01C952          4.73
0331  E  1           CALL MOVEC(OPERJ,INC,OPERA(N),IDET,NUMBA)                   00067600  01C962          4.83
0332  E  1           CALL DEBC (OPERA(N),'I',OPERA(N),4)                         00067700  01C974          6.51
0333 B               IF(ISET.EQ.DELTA(7))GO TO 1070                              00067800  01C98E          4.80
0334   L         1060 CONTINUE                                                   00067900  01C99C          6.70(  81.83) 4.8
0335 B           1070 GO TO JSUB,(470,520,550,500,320,640),890,750,              00068000  01C9B4          2.20
                *                 800,320)                                       00068100
                C                                                                00068200
                C                                                                00068300
                C                                                                00068400
                C     CONDICOES ANORMAIS                                         00068500
                C                                                                00068600
                C     ERRO TRABALHANDO NO DISCO                                  00068700
                C                                                                00068800
0336  E  2       1080 WRITE(3,1090)                                              00068900  01C98A          4.30
0337             1090 FORMAT('-',20X,'ERRO DURANTE OPERACAO DISCOS  ',//)         00069000
0338  E  1            STOP  17                                                   00069100  01C9DC          2.40
                C                                                                00069200
                C     FINAL NAO ESPERADO= IMPRIME 'SOURCE'                       00069300
                C                                                                00069400
0339           1100 CONTINUE                                                     00069500
```

```
FORTRAN IV G LEVEL  21                MAIN              DATE = 76237         10/58/11              PAGE 0013

                     C                                                                    00069600
                     C      IMPRESSAO RESTANTE INFORMACOES DO FONTE A PARTIR DA MEMORIA/DISCO 00069700
                     C                                                                    00069800
0340 B                1110 IF(NBUF.GT.ICNT) GO TO 1130                                    00069900  01C9DE            4.80
0341                       NBUF = NBUF + DELTA(7)                                         00070000  01C9FC            3.53
0342                       ICNT = ICNT - DELTA(7)                                         00070100  01C9F8            3.53
0343                       DO 1120 I=NBUF,ICNT                                            00070200  01CAC4            2.13
0344   E  1                CALL MOVE(AREA(KM),ISOUR)                                      00070300  01CACC           10.68
0345                       KM = KM + 33                                                   00070400  01CA32            3.53
0346   E  3                WRITE(3,310) ISOUR                                             00070500  01CA3E            5.50
0347    L           1120 CONTINUE                                                         00070600  01CA60            5.75(   26.39) 1.2
0348 B                1130 IF(IDIS.EQ.DELTA(3)) GO TO 1150                             00070700  01CA74            4.80
0349   E  3           1140 READ(8,END=1150) ISOUR                                       00070800  01CA82            5.50
0350   E  3                WRITE(3,310) ISOUR                                             00070900  01CAA4            4.80
0351 B                     GO TO 1140                                                     00071000  01CAC4            2.20
                     C                                                                    00071100
                     C      LE EM 'LIST'AS ESTATISTICAS DO COMPILADOR E IMPRIME           00071200
                     C                                                                    00071300
0352   E  3           1150 READ(7,40,END=1180,ERR=1080) IPR                             00071400  01CACA            5.50
                     C                                                                    00071500
                     C      VERIFICA SE HA' SUBPROGRAMAS                                  00071600
                     C                                                                    00071700
0353 B                     IF(IPRT(2).NE.CRTR)  GO TO 1170                                00071800  01CAF4            4.80
0354                       NVA  = DELTA(7)                                                00071900  01CB02            2.13
0355   E  3                READ(7,40,END=1180,ERR=1080) ILST                              00072000  01CB0A            5.50
0356   E  1                CALL MOVEC(ILST(2),2,KCH,4,1)                                  00072100  01CB34            4.83
0357 B                     IF(KCH.EQ.92) GO TO 1160                                       00072200  01CB46            6.00
0358   E  3                WRITE(3,310) IPR                                               00072300  01CB56            4.80
0359   E  1                BACKSPACE 7                                                    00072400  01CB78            2.40
0360 B                     GO TO 1180                                                     00072500  01CB84            2.20
0361   E  3           1160 WRITE(3,310) ILST                                            00072600  01CB8A            5.50
0362   E  3                READ(7,40,END=1200,ERR=1080) ILST                              00072700  01CBAC            4.80
0363 B                     IF(ILST(2).NE.CRTR) GO TO 1150                                 00072800  01CBD4            4.80
0364   E  3                WRITE(3,310) ILST                                              00072900  01CBE2            5.50
0365 B                     GO TO 1180                                                     00073000  01CC04            2.20
0366   E  3           1170 WRITE(3,310) IPR                                             00073100  01CC0A            5.50
0367 B                     GO TO 1150                                                     00073200  01CC2C            2.20
0368 B                1180 IF(NVA.EQ.DELTA(8)) GO TO 1200                              00073300  01CC32            4.80
                     C                                                                    00073400
                     C      INICIALIZA VARIAVEIS PARA PROCESSAMENTO DE SUBPROGRAMAS        00073500
                     C                                                                    00073600
0369   E  2                WRITE(3,1190)                                                  00073700  01CC40            3.60
0370                 1190 FORMAT(9X,'1234',37X,'5',3X,'6',5X,'7',7X,'8',5X,'9')         00073800
0371   E  1                REWIND 8                                                       00073900  01CC54            2.40
0372                       NBUF = DELTA(7)                                                00074000  01CC60            2.13
0373                       KM   = 34                                                      00074100  01CC68            3.33
0374                       IDIS = DELTA(8)                                                00074200  01CC74            2.13
0375                       MN   = DELTA(8)                                                00074300  01CC7C            2.13
0376                       NVA  = DELTA(8)                                                00074400  01CC84            2.13
0377                       IT   = DELTA(8)                                                00074500  01CC8C            2.13
0378                       ICNT = DELTA(8)                                                00074600  01CC94            2.13
0379                       I3   = DELTA(6)                                                00074700  01CC9C            2.13
0380                       MN = 1                                                         00074800  01CCA4            2.13
0381 B                     GO TO 30                                                       00074900  01CCAC            2.20
                     C                                                                    00075000
                     C      IMPRIME RESUMO DAS INSTRUCOES ENCONTRADAS                     00075100
                     C                                                                    00075200
0382   E  2           1200 WRITE(3,1210)                                                00075300  01CCB2            4.80
```

```
FJRTRAN IV G LEVEL  21                MAIN              DATE = 76237         10/58/11              PAGE 0014

0383          1210 FORMAT('1',2X,'RESUMO DAS INSTRUCOES JTILIZADAS NO CODIGO OBJETO   00075400
                  XGERADO PELO COMPILADOR',/,11X,'INSTRUCOES',5X,'NO.VEZES',//)       00075500
0384                DO 1230 I=1,110                                                   00075600  01CCC8              5.73
0385 8              IF(NOVEZ(I).EQ.DELTA(8)) GO TO 1230                               00075700  01CCDC              4.80
0386                NI = NI + DELTA(7)                                                00075800  01CCEA              3.53
0387  E  4          WRITE(3,1220) TAB(I),NOVEZ(I)                                     00075900  01CCF6              6.70
0388          1220 FORMAT(20X,A4,5X,I5)                                              00076000
0389    L     1230 CONTINUE                                                          00076100  01CD1C              7.90(  23.86) 4.8
0390  E  4          WRITE(3,1240) NI,TCTALG                                           00076200  01CD38              6.00
0391          1240 FORMAT('-',13X,'TOTAL= ',I3,' INSTRUCOES ',10X,'TEMPO GERAL= ',    00076300
                  *  F10.2 )                                                          00076400
0392  E  1          STOP                                                              00076500  01CD5C              2.40
0393                END                                                               00076600
```

```
FORTRAN IV G LEVEL  21                    MAIN                    DATE = 76237          10/58/11              PAGE 0015
                                      SUBPROGRAMS CALLED
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
IBCOM=      55C       MOVE        560       MOVEC       564       DEBC        568       PAREBC      56C

                                    EQUIVALENCE DATA MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
IPRT        840       IPR         840       DUMMY       840

                                        SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
MN          8C4       IDIS        8C8       KM          8CC       NBUF        8D0       ISTK        8D4
IMU         8D8       JMN         8DC       I3          8E0       NDOS        8E4       NI          8E8
IBKLE       8EC       IBALR       8F0       KCH         8F4       ICNT        8F8       IBCR        8FC
IFAS        900       J           904       I           908       NVA         90C       IT          910
NBR         914       M           918       IDL         91C       NDIG        920       LJ          924
ION         928       LOOP        92C       ASTEIX      930       ORTR        934       IBLK        938
TAM         95C       MNEN        954       TIME        958       ISTAN1      95C       ISTA        960
TTJI        964       DSTA        968       IFOR        96C       LABEL       970       IOTO        974
IDOLB       978       JSUB        97C       F1          980       IQ4         984       IR4         988
S1          98C       S3          990       S2          994       S4          998       IGR         99C
IP          9A0       IDW         9A4       ISET        9A8       JARI        9AC       N           9B0
NUMBA       9B4       INC         9B8       IDET        9BC

                                        ARRAY MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
JPERD       9C0       OPERA       9D8       LOCA        9F0       IDOLA       9FB       DELTA       A00
OLAB        A20       NOVEZ       A88       ISOUR       C40       ILST        CC4       NDG         D3C
NBRR        D4C       FPSTK       D5C       JOSTA       DC4       TEMPO       E2C       TAB         FE4
AREA       119C

                                    FORMAT STATEMENT MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
    10      1AEE9         20      1AF3F         40      18217         130      1821D         260      18248
   300      18253        310      18262       1090      18268        1190      18291        1210      18284
  1220      18323       1240      1832D

FORTRAN IV G LEVEL  21                    MAIN                    DATE = 76237          10/58/11              PAGE 0016
  *OPTIONS IN EFFECT*   NOID,EBCDIC,SOURCE,LIST,NODECK,LOAD,MAP
  *OPTIONS IN EFFECT*   NAME = MAIN     , LINECNT =       60
  *STATISTICS*     SOURCE STATEMENTS =       333,PROGRAM SIZE =   118122
  *STATISTICS*  NO DIAGNOSTICS GENERATED
```

RESUMO DAS INSTRUCOES UTILIZADAS NO CODIGO OBJETO GERADO PELO COMPILADOR

INSTRUCOES	NO. VEZES
A	35
AD	10
AE	20
AER	4
BAL	95
BALR	46
BC	13
BCR	175
BXLE	6
C	82
CE	2
CR	1
D	12
DC	236
END	1
L	544
LA	79
LCDR	10
LCR	6
LD	10
LE	49
LM	2
LPR	10
LR	4
LTR	15
M	6
ME	14
MVI	2
S	20
SE	2
SLA	26
SLL	1
SRDA	12
ST	153
STE	53
STM	2

TOTAL= 34 INSTRUCOES TEMPO GERAL= 1936.63

APÊNDICE 2

SUB-ROTINA MOVECO1

```
*                                   MOVEC 01                                    00000000
*                                                                               00000100
*                                                                               00000200
*                              MOVE CARACTERES(BYTES)                           00000300
*                                                                               00000400
*                                                                               00000500
*  ABSTRATO -ESTA SUBROTINA SIMULA A INSTRUCAO 'MVC'  DOS COMPUTADORES           00000600
*            IBM/360/370                                                        00000700
*                                                                               00000800
*                                                                               00000900
*  LINGUAGEM          - ASSEMBLER (IBM/360-370)                                 00001000
*  MEMORIA REQUERIDA  - 104 BYTES                                               00001100
*  ENTRADAS           - MOVEC OU MVC                                            00001200
*                     - MOVE                                                    00001300
*  AUTOR              - SIMPLICIO LOPES DE FREITAS        FEVEREIRO 1976        00001400
*                                                                               00001500
*  ---------------------------------------------                                00001600
*                                                                               00001700
*  ENTRADA   MOVEC OU MVC                                                       00001800
*  COMO INVOCAR       - CALL MOVEC(IDE,NIDE,IPARA,NPARA,NUMERO)                 00001900
*                 OU                                                            00002000
*                     - CALL MVC  (IDE,NIDE,IPARA,NPARA,NUMERO)                 00002100
*                       DESCRICAO E TIPO DOS ARGUMENTOS                         00002200
*                       IDE   -ENDERECO INICIAL DO ARRAY DE ONDE O DADO         00002300
*                              SERA MOVIDO                                      00002400
*                       NIDE  -NUMERO DE BYTES A PARTIR DE 'IDE' A              00002500
*                              PARTIR DO QUAL O DADO SERA MOVIDO                00002600
*                              ( 4 BYTES )                                      00002700
*                       IPARA -ENDERECO INICIAL DO ARRAY PARA ONDE O            00002800
*                              DADO SERA MOVIDO                                 00002900
*                       NPARA -NUMERO DE BYTES A PARTIR DE 'IPARA' PARA         00003000
*                              O QUAL O DADO SERA MOVIDO (4 BYTES)              00003100
*                       NUMERO-NUMERO DE BYTES(CARACT) A SEREM MOVIDOS          00003200
*                              ( 4 BYTES)                                       00003300
*                      *OBS- NIDE=1 E NPARA=1 PARA PRIMEIRO BYTE                00003400
*                                                                               00003500
*    EXEMPLO          - CALL MOVEC(IAR(5),2,IAR1(2),3,3)                       00003600
*                       TRES BYTES SAO MOVIDOS A PARTIR DO SEGUNDO              00003700
*                       BYTE DE IAR(5)  PARA  A PARTIR  DO TERCEIRO             00003800
*                       BYTE DE IAR1(2)                                         00003900
*                                                                               00004000
*  ---------------------------------------------                                00004100
*                                                                               00004200
*  ENTRADA   -MOVE                                                              00004300
*                                                                               00004400
*  FINALIDADE-MELHORAR A EFICIENCIA DO PROGRAMA DE ANALISE TEMPOS                00004500
*             ATRAVES DA NAO UTILIZACAO DE COMANDOS FORTRAN MOVENCO             00004600
*             132 BYTES DE UMA LOCACAO DE MEMORIA PARA OUTRA.                   00004700
*                                                                               00004800
*                                                                               00004900
*  COMO INVOCAR -  CALL MOVE(IENT,ISAIDA)                                       00005000
*                  DESCRICAO DOS ARGUMENTOS                                     00005100
*                  IENT  -AREA  DE ENTRADA                                      00005200
*                  ISAIDA- AREA  DE SAIDA                                       00005300
*                                                                               00005400
         LCLC  &NAM,&VER                                                       00005500
&NAM     SETC  'MOVEC'                                                         00005600
&VER     SETC  '01'                                                            00005700
         TITLE 'MOVE BYTES OU CARACTERES '                                     00005800
```

```
GNAMGVER CSECT                                                          00005900
         PRINT NOGEN                                                    00006000
         SPACE 1                                                        00006100
         ENTRY MOVEC            DESIGNA PONTOS DE ENTRADA               00006200
         ENTRY MOVE                                                     00006300
         ENTRY MVC                                                      00006400
         SPACE 2                                                        00006500
         USING *,12             DECLARA REGISTRADOR BASE                00006600
MOVE     EQU   *                                                        00006700
MVC      EQU   *                                                        00006800
MOVEC    SAVE  (14,06),T,*      FAZ SAVE DOS REGS. A SEREM USADOS       00006900
         LR    12,15            CARREGA O REG.BASE NA EXECUCAO          00007000
         ST    13,SAVE          GUARDA O REG. 13                        00007100
         SPACE 2                                                        00007200
*                                                                       00007300
         LM    2,3,0(1)         APANHA ENDERECOS DOS DOIS PRIMEIROS     00007400
*                                 PARAMETROS                            00007500
         TM    4(1),X'80'       VERIFICA SE HA MAIS PARAM = MOVEC       00007600
         BZ    MOVEC1                                                   00007700
         MVC   0(132,3),0(2)    PROCESSA A ENTRADA -MOVE-               00007800
         B     RESTORE          RETORNA                                 00007900
         SPACE 2                                                        00008000
MOVEC1   LM    4,6,8(1)         APANHA OUTROS PARAMETROS= MVC           00008100
         L     3,0(3)           R3=INCREMENTO DA ENTRADA                00008200
         BCTR  3,0              DIMINUE DE 1                            00008300
         AR    2,3              END.FINAL DA ENTRADA                    00008400
         L     5,0(5)           INCREMENTO  DA SAIDA                    00008500
         BCTR  5,0              DIMINUE DE 1                            00008600
         AR    4,5              OBTEM ENDERECO FINAL                    00008700
         L     6,0(6)           NUMERO DE BYTES A SEREM MOVIDOS         00008800
         LTR   6,6                                                      00008900
         BC    12,RESTORE                                               00009000
         BCTR  6,0              DIMINUE DE 1 PARA USAR A INSTRUCAO      00009100
         STC   6,OPMOVER+1      COLOCA NUMERO DE BYTES A SEREM          00009200
*                                 MOVIDOS NA INSTRUCAO DE MVC           00009300
OPMOVER  MVC   0(0,4),0(2)      EFETUA A OPERACAO PARA ENTRADA  MVC     00009400
*                                 OU MOVEC                              00009500
         SPACE 2                                                        00009600
RESTORE  L     13,SAVE                                                  00009700
         RETURN (14,06),T       RESTAURA REGS. E RETORNA                00009800
         SPACE 3                                                        00009900
SAVE     DS    F                                                        00010000
         END                                                            00010100
```

APÊNDICE 3

SUB-ROTINA EBCDIC

```
*                                  EBCDIC                                    00000000
*                                                                            00000100
*                                                                            00000200
*                       CONVERTE NUMERO PARA EBCDIC E VICE-VERSA             00000300
*                                                                            00000400
*                                                                            00000500
*  ABSTRATO -AS ROTINAS DE CONVERSAO DO MODULO DE EXECUCAO PARA PROGRA-       00000600
*           -MAS FORTRAN IHCFCVTH(ADCON#) SAO UTILIZADAS ATRAVES DESTA        00000700
*           SUBROTINA QUE FAZ UMA PASSAGEM  DE PARAMETROS DO USUARIO         00000800
*           PARA AQUELAS.                                                    00000900
*           OS DIFERENTES TIPOS DE CONVERSAO  ESPECIFICADOS PELOS            00001000
*           FORMATOS  'A','I','F','E','D'  PODEM SER FEITAS.                 00001100
*                                                                            00001200
*  LINGUAGEM          - ASSEMBLER (IBM/360-370)                              00001300
*  MEMORIA REQUERIDA - 340 BYTES (154 HEX)                                   00001400
*  ENTRIES            - DEBC   - CONVERTE DE EBCDIC PARA NUMERO              00001500
*                     - PAREBC - CONVERTE DE NUMERO PARA EBCDIC              00001600
*  AUTOR              - SIMPLICIO LOPES DE FREITAS        JANEIRO 1976       00001700
*                                                                            00001800
*  COMO INVOCAR       - CALL DEBC  (                        )                00001900
*                               (IENT,FORMAT,ISAI,NUM1,NUM2)                 00002000
*                       CALL PAREBC(                        )                00002100
*                                                                            00002200
*                       DESCRICAO E TIPO DOS ARGUMENTOS                      00002300
*                       IENT  -ENDERECO DO DADO A SER CONVERTIDO             00002400
*                       FORMAT- END. DE UMA PALAVRA CUJO PRIMEIRO BYTE       00002500
*                              CONTENHA A ESPECIFICACAO DE FORMAT OU          00002600
*                              LITERAL QUE INDIQUE A CONVERSAO DESEJADA       00002700
*                              ( 4 BYTES OU LITERAL )                         00002800
*                       ISAI  -ENDERECO ONDE A ENTRADA CONVERTIDA VAI        00002900
*                              SER COLOCADA.                                  00003000
*                              USANDO    A 'ENTRY' PAREBC O USUARIO DEVE      00003100
*                              PROVER QUE ESPACO SUFICIENTE SEJA ALOCADO      00003200
*                              PARA SAIDA.EXEMPLO- CONVERTENDO-SE UM NU-      00003300
*                              MERO DE 3 ALGARISMOS PARA EBCDIC  'ISAI'       00003400
*                              DEVERA TER 8 BYTES                             00003500
*                       NUM1  -ESPECIFICA O TAMANHO NA ESPECIFICACAO DO      00003600
*                              FORMATO. ( 4 BYTES )                           00003700
*                       NUM2  -NUMERO DE CASAS DECIMAIS PARA FORMATOS        00003800
*                              'F','E','D'. SO DEVE SER INCLUIDO PARA         00003900
*                              ESTES FORMATOS. ( 4 BYTES )                    00004000
*     EXEMPLOS         - CALL DEBC(ISTAN1,'I',I$TA,4)                        00004100
*                       'ISTAN1' CONTEM DADOS NUMERICOS(EM EBCDIC)           00004200
*                       QUE DEVEM SER CONVERTIDOS SEGUNDO O FORMATO 'I4'     00004300
*                       E O RESULTADO COLOCADO EM 'I$TA'                     00004400
*                      - CALL PAREBC(TMIN,'F',I$OUR(27),7,2)                 00004500
*                       OS DADOS EM 'TMIN' SAO CONVERTIDOS PARA EBCDIC       00004600
*                       SEGUNDO O FORMATO 'F7.2' E O RESULTADO E COLOCA-     00004700
*                       -DO A PARTIR DE 'I$OUR(27)'                          00004800
*                                                                            00004900
         EJECT                                                              00005000
EBCDIC   CSECT 0                                                            00005100
         PRINT NOGEN                                                        00005200
         ENTRY DEBC              DEFINE AS 2                                00005300
         ENTRY PAREBC               ENTRIES                                 00005400
         EXTRN ADCON#            DECLARA REFERENCIA EXTERNA                 00005500
         USING *,15              DECLARA REG.BASE                           00005600
DEBC     STM   14,12,12(13)      FAZ O SAVE DOS REGISTRADORES               00005700
         LA    11,0              FAZ R11=0 PARA ESTA ENTRY                  00005800
```

```
         L      3,0(1)            R3=END.ENTRADA(IENT)                    00005900
         L      2,8(1)            R2=END.SAIDA  (ISAI)                    00006000
         B      INICIO            VAI PARA O PROCESSAMENTO COMUM AS       00006100
*                                   DUAS ENTRIES                          00006200
         DROP   15                                                        00006300
         USING *,15               DECLARA REG.BASE                        00006400
PAREBC   STM    14,12,12(13)      FAZ O SAVE DOS REGISTADORES             00006500
         LA     11,4              FAZ R11=4 PARA ESTA ENTRY               00006600
         L      2,0(1)            R2=END.ENTRADA(IENT)                    00006700
         L     3,8(1)             R3=END.SAIDA  (ISAI)                    00006800
         B      INICIO                                                    00006900
         DROP   15                                                        00007000
*                                                                         00007100
* PROCESSAMENTO  COMUM.AS 2 ENTRIES                                       00007200
*                                                                         00007300
INICIO   BALR   12,0               R12 E O REG.BASE                      00007400
         USING  *,12                                                     00007500
*                                                                         00007600
* FAZ CONVENCOES DE LINKAGE E 'SAVE' ADICIONAL                            00007700
*                                                                         00007800
         LR     7,13              R7=CONTEM END.AREA SAVE                 00007900
         LA     13,SAVE           R13=END.NOVA AREA SAVE                  00008000
         ST     13,8(0,7)                                                 00008100
         ST     7,4(0,13)                                                 00008200
*                                                                         00008300
         LR     8,1               R8=ENDERECO DA LISTA DE PARAMETROS      00008400
         L      1,=A(ADCON#)      R1=END.DAS ROTINAS DE CONVERSAO         00008500
         L      4,4(8)            R4=ENDERECO DE FORMAT                   00008600
         L      9,12(8)           R9=ENDERECO DE NUM1                     00008700
         L      9,0(9)            R9=VALOR DE NUM1                        00008800
         LA     6,0               R6=0                                    00008900
         STH    6,CONH              INICIALIZA A AREA QUE OS PARAMETROS   00009000
         STH    6,CONF                        SERAO PASSADOS PARA       00009100
         STH    6,CONF+2                           ADCON=                 00009200
*                                                                         00009300
         CLI    0(4),C'I'         VERIFICA SE FORMATO PASSADO 'I'         00009400
         BE     I                 SE FOR IGUAL 'I' VAI PARA 'I'           00009500
         CLI    0(4),C'A'         VERIFICA SE FORMATO E 'A'               00009600
         BE     A                                                         00009700
*                                                                         00009800
* FORMATO NAO E  'I' OU 'A' APANHA PARAMETRO NUM2                         00009900
*                                                                         00010000
         L      10,16(8)          R10=ENDERECO(NUM2)                      00010100
         L      10,0(10)          R10=VALOR NUM2                          00010200
         CLI    0(4),C'F'         VERIFICA SE FORMATO 'F'                 00010300
         BE     F                                                         00010400
         CLI    0(4),C'E'         VERIFICA SE FORMATO 'E'                 00010500
         BE     E                                                         00010600
         CLI    0(4),C'D'         VERIFICA SE FORMATO 'D'                 00010700
         BE     D                                                         00010800
*                                                                         00010900
I        STC    9,CONH+1          WW = NUM1                               00011000
         MVI    CONH,X'04'        LL                                      00011100
         L      1,40(11,1)        R1=END.ROTINA CONVERSAO 'I'             00011200
         B      AI                                                        00011300
*                                                                         00011400
*  PARA 'A','F','E','D' FAZ O MESMO PROCEDIMENTO QUE PARA 'I'             00011500
*                                                                         00011600
A        STC    9,CONH+1                                                  00011700
```

```
         STC   9,CONH                                                  00011800
         L     1,80(11,1)                                              00011900
         B     AI                                                      00012000
F        MVI   CONF,X'04'                                              00012100
         STC   9,CONF+1                                                00012200
         STC   10,CONF+2          DD - NUM2                            00012300
         L     1,0(11,1)                                               00012400
         B     FED                                                     00012500
E        MVI   CONF,X'04'                                              00012600
         STC   9,CONF+1                                                00012700
         STC   10,CONF+2                                               00012800
         L     1,8(11,1)                                               00012900
         B     FED                                                     00013000
D        MVI   CONF,X'C8'                                              00013100
         STC   9,CONF+1                                                00013200
         STC   10,CONF+2                                               00013300
         L     1,8(11,1)                                               00013400
         B     FED                                                     00013500
*                                                                      00013600
*  PARA CONVERSOES 'A' E 'I' OS PARAMETROS PASSADOS A ADCON= SAO       00013700
*        DC XL2'LLWW' AONDE LL=TAMANHO EM BYTES DA SAIDA               00013800
*                           WW=TAMANHO EM BYTES DA ENTRADA             00013900
*                                                                      00014000
AI       BALR  0,1                VAI PARA A ROTINA DE CONVERSAO      00014100
CONH     DS    XL2                'LLWW'                              00014200
         B     FINAL                                                   00014300
*                                                                      00014400
*  PARA CONVERSOES 'F','E','D' OS PARAMETROS SAO                       00014500
*        DC    XL4'LLWWDDYY' AONDE                                     00014600
*                           DD=NUMERO DE CASAS DECIMAIS                00014700
*                           YY=NAO USADA(ZERADA)                       00014800
*                                                                      00014900
FED      BALR  0,1                VAI PARA A ROTINA DE CONVERSAO      00015000
CONF     DS    XL4                'LLWWDDYY'                          00015100
*                                                                      00015200
*  CONVENCOES DE RETORNO                                               00015300
*                                                                      00015400
FINAL    L     13,SAVE+4                                               00015500
         RETURN (14,12),T                                              00015600
         DS    0F                 REQUER ALINHAMENTO DE PALAVRA       00015700
SAVE     DS    18F                AREA DE SAVE DESTE SUBPROGRAMA      00015800
         END                                                           00015900
```

APÊNDICE 4

PROCEDIMENTOS CATALOGADOS

```
//FORTTC  PROC  PCLASS=DUMMY,TRI=200,PGMNAME=ANATEMP                    00000000
//FORT    EXEC PGM=IEYFORT,REGION=180K,PARM=LIST                        00000100
//SYSPRINT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&SIMP,                     00000200
//            SPACE=(TRK,(&TRI,10)),                                    00000300
//            DCB=(LRECL=120,RECFM=FBSA,BLKSIZE=7200)                   00000400
//SYSPUNCH DD &PCLASS                                                   00000500
//SYSLIN   DD DSN=&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,                  00000600
//            SPACE=(CYL,(2,2)),DCB=BLKSIZE=800                         00000700
//*                                                                     00000800
//TIME    EXEC PGM=&PGMNAME,REGION=180K                                 00000900
//STEPLIB  DD DSN=PIS.LOADLIB,DISP=SHR                                  00001000
//FT01F001 DD DDNAME=SYSIN                                              00001100
//FT03F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=7132)           00001200
//SYSPRINT DD SYSOUT=A                                                  00001300
//FT07F001 DD DSN=&SIMP,DISP=(OLD,DELETE,DELETE)                        00001400
//FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),DSN=&ABC,             00001500
//            SPACE=(TRK,(&TRI,10)),                                    00001600
//            DCB=(LRECL=132,RECFM=VSBA,BLKSIZE=7260)                   00001700
//*                                                                     00001800
```

```
//FORTTCL PROC  PCLASS=DUMMY,TRI=200,PGMNAME=ANATEMP                    00000000
//FORT    EXEC PGM=IEYFORT,REGION=180K,PARM=LIST                        00000100
//SYSPRINT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&SIMP,                     00000200
//            SPACE=(TRK,(&TRI,10)),                                    00000300
//            DCB=(LRECL=120,RECFM=FBSA,BLKSIZE=7200)                   00000400
//SYSPUNCH DD &PCLASS                                                   00000500
//SYSLIN   DD DSN=&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,                  00000600
//            SPACE=(CYL,(2,2)),DCB=BLKSIZE=800                         00000700
//*                                                                     00000800
//TIME    EXEC PGM=&PGMNAME,REGION=180K                                 00000900
//STEPLIB  DD DSN=PIS.LOADLIB,DISP=SHR                                  00001000
//FT01F001 DD DDNAME=SYSIN                                              00001100
//FT03F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=7132)           00001200
//SYSPRINT DD SYSOUT=A                                                  00001300
//FT07F001 DD DSN=&SIMP,DISP=(OLD,DELETE,DELETE)                        00001400
//FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),DSN=&ABC,             00001500
//            SPACE=(TRK,(&TRI,10)),                                    00001600
//            DCB=(LRECL=132,RECFM=VSBA,BLKSIZE=7260)                   00001700
//*                                                                     00001800
//LKED    EXEC PGM=IEWL,REGION=180K,PARM=(XREF,LET,LIST),               00001900
//            COND=(4,LT,FORT)                                          00002000
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR                                 00002100
//         DD DSN=WGC.SUBRLIB,DISP=SHR                                  00002200
//SYSLMOD  DD DSN=&GOSET(MAIN),DISP=(,PASS),UNIT=SYSDA,                 00002300
//            SPACE=(CYL,(1,1,1))                                       00002400
//SYSPRINT DD SYSOUT=A                                                  00002500
//SYSUT1   DD DSN=&SYSUT1,SPACE=(CYL,(2,2)),UNIT=SYSDA                 00002600
//SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)                            00002700
//         DD DDNAME=SYSIN                                              00002800
//*                                                                     00002900
```

```
//FORTTCLG PROC PCLASS=DUMMY,TRI=200,PGMNAME=ANATEMP                    00000000
//FORT    EXEC PGM=IEYFORT,REGION=180K,PARM=LIST                        00000100
//SYSPRINT DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&SIMP,                     00000200
//            SPACE=(TRK,(&TRI,10)),                                    00000300
//            DCB=(LRECL=120,RECFM=FBSA,BLKSIZE=7200)                   00000400
//SYSPUNCH DD &PCLASS                                                   00000500
//SYSLIN   DD DSN=&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,                  00000600
//            SPACE=(CYL,(2,2)),DCB=BLKSIZE=800                         00000700
//*                                                                     00000800
//TIME    EXEC PGM=&PGMNAME,REGION=180K                                 00000900
//STEPLIB  DD DSN=PIS.LOADLIB,DISP=SHR                                  00001000
//FT01F001 DD DDNAME=SYSIN                                              00001100
//FT03F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=7132)           00001200
//SYSPRINT DD SYSOUT=A                                                  00001300
//FT07F001 DD DSN=&SIMP,DISP=(OLD,DELETE,DELETE)                        00001400
//FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),DSN=&ABC,             00001500
//            SPACE=(TRK,(&TRI,10)),                                    00001600
//            DCB=(LRECL=132,RECFM=VSBA,BLKSIZE=7260)                   00001700
//*                                                                     00001800
//LKED    EXEC PGM=IEWL,REGION=180K,PARM=(XREF,LET,LIST),               00001900
//            COND=(4,LT,FORT)                                          00002000
//SYSLIB   DD DSN=SYS1.FORTLIB,DISP=SHR                                 00002100
//         DD DSN=WGC.SUBRLIB,DISP=SHR                                  00002200
//SYSLMOD  DD DSN=&GOSET(MAIN),DISP=(,PASS),UNIT=SYSDA,                 00002300
//            SPACE=(CYL,(1,1,1))                                       00002400
//SYSPRINT DD SYSOUT=A                                                  00002500
//SYSUT1   DD DSN=&SYSUT1,SPACE=(CYL,(2,2)),UNIT=SYSDA                  00002600
//SYSLIN   DD DSN=&LOADSET,DISP=(OLD,DELETE)                            00002700
//         DD DDNAME=SYSIN                                              00002800
//*                                                                     00002900
//GO      EXEC PGM=*.LKED.SYSLMOD,COND=((4,LT,FORT),(4,LT,LKED))       00003000
//FT01F001 DD DDNAME=SYSIN                                              00003100
//FT03F001 DD SYSOUT=A,DCB=(RECFM=UA,BLKSIZE=133)                       00003200
//FT02F001 DD SYSOUT=B                                                  00003300
```

APÊNDICE 5

SUB-ROTINA TIME01

```
*                                TIME01                                 00000000
*                                                                       00000100
*                                                                       00000200
*                     MEDICAO DE TEMPO DE EXECUCAO                      00000300
*                                                                       00000400
*                                                                       00000500
*  ABSTRATO -ESTA SUBROTINA FAZ USO DAS MACROS DE 'OS-MVT' STIMER  E     00000600
*            TTIMER.                                                    00000700
*            DA AO PROGRAMADOR A FACILIDADE DE MEDIR O TEMPO DE EXECU-  00000800
*            -CAO DE TRECHOS PREVIAMENTE ESCOLHIDOS DENTRO DE UM        00000900
*            PROGRAMA                                                   00001000
*                                                                       00001100
*  LINGUAGEM          - ASSEMBLER (IBM/360-370)                         00001200
*  MEMORIA REQUERIDA  - 508 BYTES(1FC HEX)                              00001300
*  ENTRY              - TIME                                            00001400
*  AUTOR              - SIMPLICIO L.DE FREITAS          JANEIRO 1976    00001500
*                                                                       00001600
*  COMO INVOCAR       - CALL TIME                                       00001700
*                       A PRIMEIRA CHAMADA A TIME FORNECE O TEMPO MAXIMO 00001800
*                       QUE PODE SER MEDIDO ATRAVES DESTA SUBROTINA.    00001900
*                       CHAMADAS SUBSEQUENTES FORNECEM O TEMPO ENTRE A  00002000
*                       ULTIMA CHAMADA E A ATUAL.                       00002100
*                                                                       00002200
*  PRECISAO DOS       - DESVIOS AINDA MAIORES DO QUE A RESOLUCAO  DO    00002300
*   RESULTADOS          'TIMER'(16.7 MSEG.P/FREQ.60HZ) SAO POSSIVEIS    00002400
*                       DEPENDENDO DO NUMERO DE VEZES EM QUE O PROGRA-  00002500
*                       -MA UTILIZANDO ESTA  SUBROTINA  SEJA INTERROM-  00002600
*                       -PIDO EM UM SISTEMA DE MULTIPROGRAMACAO         00002700
*                                                                       00002800
         EJECT                                                          00002900
TIME01   CSECT                                                          00003000
         PRINT NOGEN              MACROS NAO SERAO EXPANDIDAS           00003100
         ENTRY TIME               DECLARA ENTRY                         00003200
         USING *,2                DECLARA O REG.BASE                    00003300
         USING IHADCB,11                                                00003400
TIME     EQU   *                                                        00003500
*                                                                       00003600
*                    CONVENCOES DE LINKAGE                              00003700
         SAVE  (14,12),T,*                                              00003800
         LR    2,15                                                     00003900
         LA    15,AREASAVE                                              00004000
         ST    15,8(13)                                                 00004100
         ST    13,AREASAVE+4                                            00004200
         LR    13,15                                                    00004300
*                                                                       00004400
         TTIMER CANCEL            A MACRO TTIMER FAZ O PROGRAMA DE      00004500
*                                  CONTROLE RETORNAR NO REG.0 O TEMPO    00004600
*                                  QUE SOBROU DO INTERVALO ANTES FIXADO  00004700
*                                  PELA MACRO STIMER.                    00004800
         L     5,TUINTVIR         R5=50000000                           00004900
         SR    5,0                SUBTRAI DE R0 E OBTEM QUANTO GASTOU   00005000
         LR    7,5                R7=R5                                 00005100
*                                                                       00005200
*  CADA UNIDADE DEVE SER MULTIPLICADA POR 1/60=26.04166 MICROSEGUNDOS.  00005300
*  EH FEITO UM ARREDONDAMENTO PARA 26.042                               00005400
*                                                                       00005500
         M     4,=F'26'           MULTIPLICA R5 POR 26                  00005600
         M     6,=F'42'           MULTIPLICA R7 POR 42                  00005700
         SR    6,6                R6=0                                  00005800
```

```
         D     6,=F'1000'          DIVIDE R7 POR 1000                        00005900
         AR    5,7                 OBTEM TOTAL                               00006000
*                                                                            00006100
         TM    DCBOFLGS,X'10'      ARQUIVO IMPRESSAO                         00006200
         BO    NAOABRE                          ESTA  ABERTO                 00006300
         OPEN  (PRINTER,(OUTPUT)) ABRE O ARQUIVO DE SAICA PARA IMPR.         00006400
NAOABRE  CVD   5,DECOMP            CONVERTE P/ DECIMAL O VALOR TOTAL         00006500
         MVC   SAIDA,SAIDA-1       LIMPA A AREA DE SAIDA                     00006600
         MVI   SAIDA,C'0'          SALTA 2 LINHAS                            00006700
         MVC   SAIDA+10(13),PADRAO    FAZ A EDICAO DA RESPOSTA               00006800
         ED    SAIDA+10(13),DECOMP+2   DE ACORDO COM PADRAO                  00006900
         MVC   SAIDA+27(10),SEGUNDO   COLOCA 'SEGUNDOS' NA SAIDA             00007000
         PUT   PRINTER,SAIDA         FAZ A IMPRESSAO                        00007100
*                                                                            00007200
*                                                                            00007300
SYMBOL   STIMER TASK,TUINTVL=TUINTVIR  CAUSA AO PROGRAMA CONTROLE A          00007400
*                                    TER UM 'TIMER' PROGRAMADO (COM         00007500
*                                    O TEMPO DEF.EM TUINTVL,E INDICA        00007600
*                                    QUE ESTE VALOR SO SERA DECREMEN-       00007700
*                                    -TADO QUANDO A 'TASK' PRESENTE         00007800
*                                    ESTIVER ATIVA                          00007900
         L     13,AREASAVE+4          LINKAGE DE RETORNO                    00008000
         RETURN (14,12),T               DESTA SUBROTINA                     00008100
*                                                                            00008200
*  DESCRICAO DO ARQUIVO DE SAIDA                                             00008300
PRINTER  DCB  DDNAME=FT03F001,DSORG=PS,MACRF=PM,RECFM=UA,BLKSIZE=133         00008400
*                                                                            00008500
         DS    0D                    ALINHAMENTO EM PALAVRA DUPLA          00008600
AREASAVE DC    18F'0'                AREA DE SAVE DESTA SUBROTINA          00008700
TUINTVIR DC    F'50000000'           UNIDADES DE TEMPO ,                   00008800
         DC    CL1' '                                                      00008900
SAIDA    DC    133C' '               AREA DE SAIDA PARA IMPRESSAO          00009000
DECOMP   DS    D                     AREA TRABALHO PARA CONVERSAO          00009100
PADRAO   DC    X'40202020202168202020202020'                               00009200
SEGUNDO  DC    CL10' SEGUNDOS '                                            00009300
         DS    0D                                                          00009400
         DCBD  DSORG=PS                                                    00009500
         END                                                               00009600
```

REFERÊNCIAS BIBLIOGRÁFICAS

A) LIVROS E MANUAIS

|1| IBM - System 360, Operating System, Fortran IV (G and H) Programmer's Guide, GC28-6817-4, USA, 1973.

|2| KNUTH, Donald E. - The Art of Computer Programming, vol.1, Fundamental Algorithms, Adison - Wesley Publishing Company, USA, 1973.

|3| HEHL, Maximilian Emil - Fortran: Técnicas Práticas e Eficientes em Programação, Livros Técnicos e Científicos Editora S/A, Rio de Janeiro, 1974.

|4| IBM, Fortran IV (H) Compiler, Program Logic Manual, GY28-6642-5, USA, 1972.

|5| IBM, System 360/model 65, Functional Characteristics, GA22-6884-4, USA, 1971.

|6| IBM, OS SMF, GC28-6712-7, USA, 1973.

|7| IBM, OS Release 21, Supervisor Services and Macro Instructions, GC28-6646-7, USA, 1974.

|8| IBM, System/370 Model 165, Functional Characteristics, GA22-6935-0, USA, 1970.

|9| IBM, OS Fortran IV (H Extended) Compiler - Programmer's Guide, SC28-6852-1, USA, 1972.

B) ARTIGOS DE REVISTAS

|10|HATFIELD D.J. e GERALD J. - Program Restructuring for Virtual Memory, IBM Systems Journal - Vol.X - nº 3, USA, 1971.

|11| SHEDLER, G.S. e YANG S. C. - Simulation of a Model of Paging System Performance, IBM System Journal - vol. X - nº 2, USA, 1971.

|12| ROGERS, J.G. - Structured Programming for Virtual Storage, IBM Systems Journal - Vol.XIV - nº IV, USA; 1975.