

**Universidade de São Paulo  
USP  
São Carlos**

**Descobrendo  
o ambiente do  
FORTRAN  
(Versão 5.1)**

**Marcelo Augusto Santos Turine  
José Carlos Maldonado  
Rosely Sanches**

**Agosto de 1995**

## 1. Introdução

Este trabalho é destinado a estudantes de graduação interessados em familiarizar-se com o compilador Fortran (versão 5.1) do ambiente da Microsoft. A linguagem FORTRAN nasceu em 1956 e foi a primeira linguagem de programação de alto nível a ser proposta. Foi sugerida visando a resolução de problemas da área científica através do uso de computadores. Seu nome é uma composição de **FOR**mula **TRAN**slation.

Basicamente, um programa é uma seqüência de comandos que descrevem as ações que você deseja que o computador realize, ou seja, um programa consiste na tradução de um algoritmo em uma forma inteligível para a máquina, o que é feito com o uso de uma linguagem de programação. Um programa descrito/especificado em uma linguagem de alto nível é denominado de *programa fonte*. Um *algoritmo* é uma seqüência ordenada de passos executáveis e precisamente definidos, que manipulam um conjunto de informações a fim de obter um resultado. Estas informações são denominadas de dados [Cereda & Maldonado, 1987].

Para que um programa possa ser executado é necessário cumprir algumas etapas (ver Figura 1):

♦ **Compilação:** Esta etapa traduz o programa fonte para o programa objeto, que são arquivos de código objeto com a extensão “.OBJ”. Esses arquivos contêm um código binário que não pode ser entendido na exibição e nem na impressão.

♦ **Composição:** Ao ser chamado, o editor de ligação (*linker*) toma um ou mais arquivos “.OBJ” e processa-os juntos para produzir um único arquivo executável com a extensão “.EXE”. O editor de ligação pode também processar o código proveniente de bibliotecas de rotinas pré-compiladas fornecidas pelo Fortran.

♦ **Execução:** Após criar o arquivo executável, seu programa já pode ser executado. Depois do comando de execução (o nome do arquivo executável) entre com os dados exigidos pelo programa (se necessários) e espere os resultados.

Os programas fontes Fortran devem sempre ser identificados através de um nome, que deverá conter, obrigatoriamente, a extensão “.FOR”. As etapas de compilação e composição descritas acima gerarão dois novos arquivos identificados pelo mesmo nome atribuído ao arquivo programa fonte, porém diferenciados pelas extensões “.OBJ” e “.EXE”, respectivamente (por exemplo: *primeiro.for*, *primeiro.obj*, *primeiro.exe*).

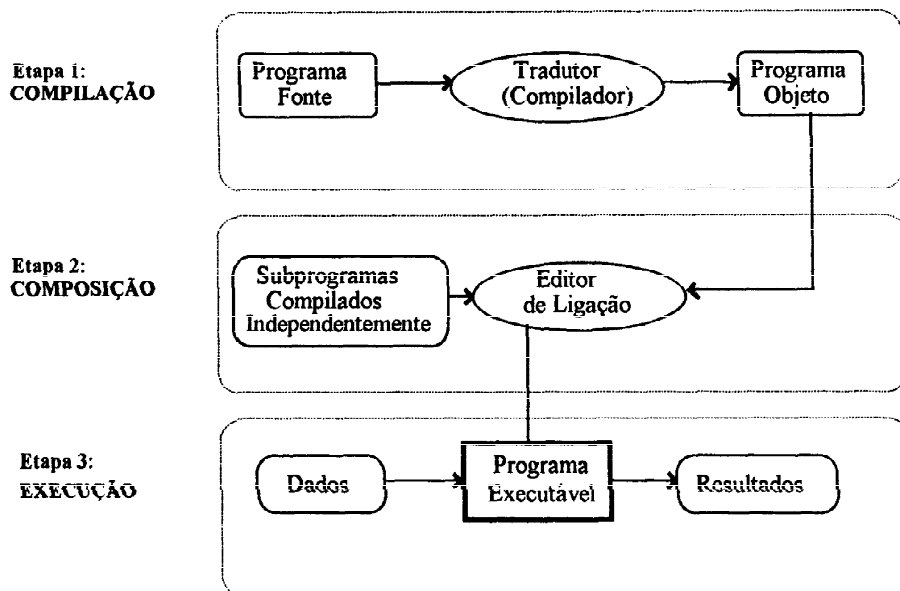


Figura 1: Etapas de um processamento.

O objetivo deste trabalho é apresentar as características gerais do compilador Fortran (versão 5.1) do ambiente da Microsoft, sem entrar em detalhes ou particularidades. Na seção 2 é apresentado o ambiente do Fortran, e a seguir é criado um pequeno programa Fortran para exemplificar as etapas descritas acima. As etapas gerais de processamento de um programa, bem como, algumas situações de erros que podem ocorrer nessas etapas e os possíveis procedimentos de correção destes erros são apresentados nas seções 3 e 4. Por último, são apresentadas as conclusões do trabalho.

## 2. Ambiente Fortran

Nesta seção, inicialmente, são apresentados os passos necessários para entrar no ambiente Fortran. É importante ressaltar que tais passos são válidos para os microcomputadores configurados no Centro de Informática de São Carlos (CISC — USP, São Carlos). Observe que estes passos devem ser adaptados de acordo com a configuração do ambiente Fortran que o usuário trabalha.

Após ligar o microcomputador, siga os passos abaixo para entrar no ambiente Fortran a partir do sistema operacional DOS (*Disk Operating System*):

1. Entre no diretório "D". Para isto, digite "D: <Enter>" após o "prompt" do DOS.

C:\>D: <Ente>      =>      D:\>

2. Entre no diretório que contém os arquivos necessários ao Fortran. Por exemplo, se o Fortran está instalado no diretório "FORTRAN" digite "*cd FORTRAN <Enter>*".

**D:\>*cd FORTRAN <Enter>* ⇒ D:\FORTRAN\>**

Se você instalou o Fortran em outro diretório, faça o mesmo, dando o nome do diretório e o disco.

3. Uma vez no diretório do Fortran, execute o arquivo "forbat2.bat" que contém os comandos essenciais para configurar o ambiente.

**D:\FORTRAN\>*forbat2 <Enter>***

Se você não encontrou o arquivo "forbat2.bat" no diretório do Fortran é necessário executar os passos abaixo. Atenção, estes passos supõe-se que você instalou os arquivos do fortran em "D:\FORTRAN" e que "C:\TMP" é o diretório de trabalho. Altere estes nomes se necessário.

```
SET PATH=D:\FORTRAN\BIN; D:\FORTRAN\BINB;  
D:\FORTRAN\LIB  
SET LIB=D:\FORTRAN\LIB  
SET INCLUDE=D:\FORTRAN\INCLUDE  
SET HELPFILES=D:\FORTRAN\HELP\*.HLP  
SET INIT=CATMP  
SET TMP=CATMP  
COPY D:\FORTRAN\CURRENT.STS C:\TMP
```

4. Finalmente, digite "*pwb*". Este comando é obrigatório e independe da instalação do Fortran. "PWB" vem de **P**rogrammer's **W**ork**B**ench — uma ferramenta abrangente para o desenvolvimento de aplicações.

**D:\FORTRAN>*pwb <Enter>***

Em seguida, aparecerá a janela principal do ambiente Fortran, ilustrada na Figura 2 [Environment and Tools, 1991; Advanced Topics, 1991].

A janela principal possui os seguintes componentes:

♦ **Menu Bar** (Menu Principal): corresponde à linha superior da janela e contém a lista das principais opções do ambiente.

♦ **Janela de Edição**: contém o programa fonte ou exibe informações a respeito da compilação e do sistema de ajuda (help).

♦ **Minimize/Maximize button** (Botão de Minimizar/Maximizar): botão que selecionado maximiza a janela (preenche a tela inteira) ou restaura o tamanho da janela para o seu tamanho original.

♦ **Title Bar** (Barra de Título): apresenta o nome do arquivo que está na janela de edição.

♦ **Scroll Bars**: modifica a posição no arquivo.

♦ **Status Bar** (Barra de Status): apresenta as opções para ajuda, resume o conteúdo do menu ou exibe outras informações.

♦ **File type indicator** (Indicador do tipo do arquivo): apresenta o tipo do arquivo presente na janela de edição. Os tipos de arquivo são:

- fortran — código fonte Fortran.
- C — código fonte C.
- cobol — código fonte Cobol.
- assembly — código fonte na linguagem Assembly.
- basic — código fonte Basic.

♦ **Line and column indicators** (Indicadores de linha e coluna): apresenta a linha e coluna nas quais o cursor está posicionado na janela de edição

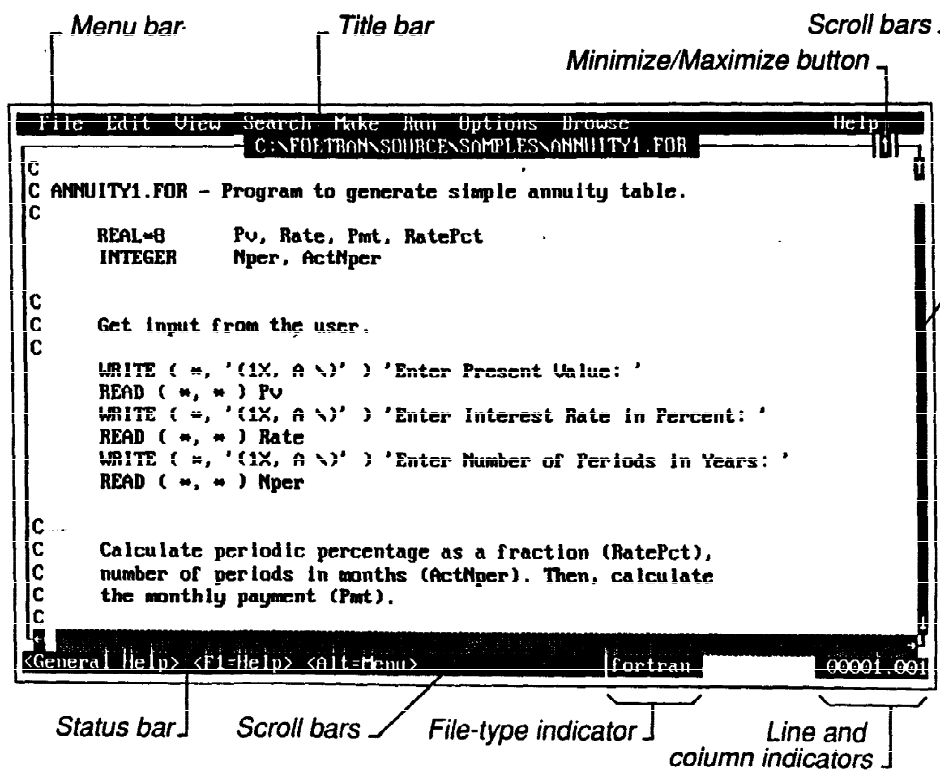
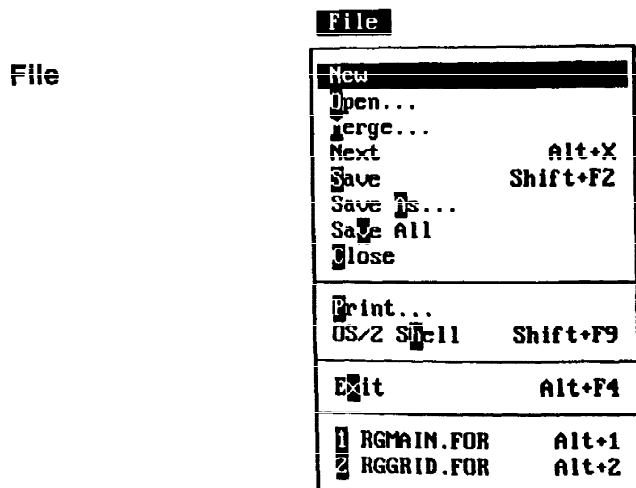


Figura 2: Janela Principal do ambiente fortran.

Se o *mouse* estiver instalado basta selecionar, diretamente, as opções do menu principal. Caso contrário, para ativar o menu principal utiliza-se as seguintes teclas:

- A tecla “*Alt*” leva você da janela de edição para o menu principal.
- “*Esc*” leva você de volta para a janela de edição.
- “*Alt + Letra*” oferece uma seleção do menu principal em qualquer instante.

A seguir, explicaremos cada uma das opções do menu principal. Caso você já esteja familiarizado com estas opções vá diretamente à seção 3.



- New:** Iniciar a edição de um novo arquivo.
- Open:** Abrir um arquivo já criado anteriormente.
- Merge:** Concatenar um ou mais arquivos dentro do arquivo atual.
- Next:** Exibir o próximo arquivo da lista de arquivos.
- Save:** Salvar o arquivo atual.
- Save As:** Salvar o arquivo atual com nome diferente.
- Save All:** Salvar todos os arquivos modificados.
- Close:** Fechar arquivo atual.
- Print:** Imprimir o arquivo atual ou o que está selecionado no arquivo.
- DOS Shell:** Sair temporariamente para o DOS (retorna digitando “Exit”).
- Exit:** Sair do ambiente Fortran.

## Edit

Edit	
Undo	Alt+Bksp
Redo	
Repeat	
Cu	Shift+Del
Copy	Ctrl+Ins
Paste	Shift+Ins
Clear	Del
Set Anchor	
Select To Anchor	
Box Mode	
Read Only	
Set Record...	
Record On	Shift+Ctrl+R
Edit Macro	

Undo:	Inverter o efeito de sua operação de edição mais recente.
Redo:	Inverter o efeito do último comando Undo.
Repeat:	Repetir a última operação de edição.
Cut:	Eliminar um bloco de texto selecionado na janela de edição e copiar para o "clipboard" (área de transferência).
Copy:	Copiar um bloco de texto selecionado para o clipboard.
Paste:	Inserir um bloco de texto do "clipboard" na janela de edição.
Clear:	Limpar o texto selecionado sem copiar para o clipboard.
Set Anchor:	Salvar a posição atual do cursor como referência para a seleção de texto.
Select To Anchor:	Selecionar o texto marcado até a posição atual do cursor.
Box Mode:	Modificar o modo de seleção do texto (Box Mode, Line Mode, Stream Mode)
Read Only:	Proteger o arquivo deixando-o somente para leitura.
Set Record:	Definir um nome e uma chave (tecla) para uma macro, ou seja, um comando que executa um conjunto de comandos a fim de desempenhar uma determinada função.
Record On:	Registrar a macro para ser editada.
Edit Macro:	Editar uma macro registrada.

Além das opções apresentadas acima, os comandos básicos que compõem a etapa de edição do programa são:

### ***Controle e Avanço do Cursor:***

<b>Seta para cima (↑):</b>	Faz o cursor subir uma linha.
<b>Seta para baixo (↓):</b>	Faz o cursor descer uma linha.
<b>Seta para a esquerda (←):</b>	Faz o cursor se mover um espaço para a esquerda.
<b>Seta para a direita (→):</b>	Faz o cursor se mover um espaço para a direita.
<b>Home:</b>	Faz o cursor se mover para o início da linha.
<b>End:</b>	Faz o cursor se mover para o final da linha.
<b>PgUp:</b>	Faz aparecer a página anterior.
<b>PgDown:</b>	Faz aparecer a página seguinte.
<b>Ctrl-A:</b>	Movimenta o cursor para a esquerda, uma palavra inteira de cada vez.
<b>Ctrl-F:</b>	Movimenta o cursor para a direita, uma palavra inteira de cada vez.
<b>Ctrl Home:</b>	Movimenta o cursor para a primeira linha do texto.
<b>Ctrl End:</b>	Movimenta o cursor para a última linha do texto.

### ***Inserção e Remoção de texto:***

<b>Ins:</b>	Muda o modo de Inserir para Sobrescrever e vice-versa.
<b>Retrocesso:</b>	Remove o caractere à esquerda do cursor e preenche o espaço com os caracteres que estão à direita do cursor.
<b>Del:</b>	Remove o caractere na posição do cursor e preenche o espaço com os caracteres que estão à direita do cursor.
<b>Ctrl Y:</b>	Remove uma linha inteira.



## View

View	
Split Horizontal	
Split Vertical	
Size Window	Ctrl+F8
Maximize Window	Ctrl+F10
Close Window	Ctrl+F4
Compile Results	

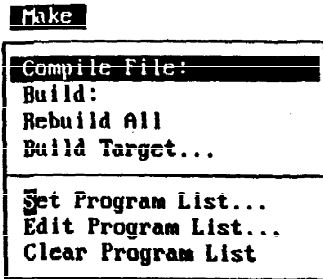
Split Horizontal:	Dividir a janela de edição horizontalmente.
Split Vertical:	Dividir a janela de edição verticalmente.
Size Window:	Modificar o tamanho da janela de edição.
Maximize Window:	Maximizar a janela de edição.
Close Window:	Fechar a janela de edição.
Compile Results:	Mostrar os resultados da última compilação na janela de edição.

## Search

Search	
Find...	
Selected Text	F3
Repeat Last Find	F3
Change...	
For File...	
Next Error	Shift+F3
Previous Error	Shift+F4
Set Error	
Go To Mark...	
Define Mark...	
Set Mark File...	

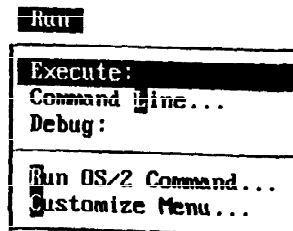
Find:	Buscar a ocorrência de uma palavra no arquivo.
Selected Text:	Buscar a próxima ocorrência do texto marcado na janela de edição.
Repeat Last Find:	Repetir a última busca.
Change:	Buscar uma palavra e trocar por outra.
For File:	Buscar um arquivo em disco.
Next Error:	Mover para o texto que ocorreu erro.
Previous Error:	Mover para o texto que ocorreu o erro anterior.
Set Error:	Selecionar o erro e a sua posição no texto.
Go to Mark:	Mover o cursor para a posição marcada no arquivo.
Define Mark:	Criar uma marca associando um nome com a posição no arquivo.
Set Mark File:	Criar um arquivo contendo todas as marcas definidas.

## Make



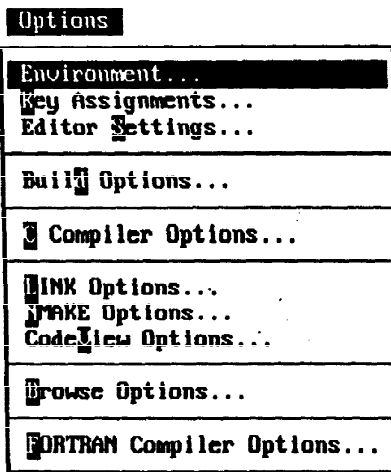
Compile File:	Compilar o arquivo atual.
Build:	Compilar e passar no editor de ligação todos os arquivos modificados.
Rebuild All:	Compilar e compor todos os arquivos.
Build Target:	Associar a lista de programa a um arquivo.
Set Program List:	Criar ou abrir uma lista de programa que define os arquivos componentes do programa.
Edit Program List:	Modificar o conteúdo da lista de programa.
Clear Program List:	Remover a lista de programa atual da memória, sem modificar o conteúdo.

## Run



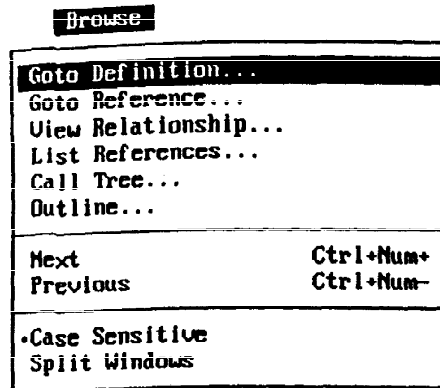
Execute:	Executar o programa atual.
Command Line:	Especificar os argumentos da linha de comando para executar o programa.
Debug:	Invocar o Depurador Microsoft CodeView.
Run DOS Command:	Executar comandos do DOS sem sair do ambiente do Fortran.
Customize Menu:	Adicionar comandos do usuário neste menu (Run).

## Options



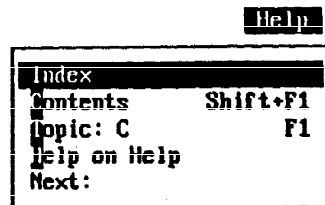
Environment:	Especificar os caminhos para os arquivos necessários de includes, bibliotecas e ajuda online.
Key Assignments:	Especificar palavras chaves que invocam comandos e macros.
Editor Settings:	Modificar a configuração do editor do Fortran.
Build Options:	Escolher a linguagem principal e outros parâmetros.
Compiler Options:	Escolher as opções de compilação.
Link Options:	Escolher as opções do editor de ligação.
NMake Options:	Escolher a linha de comando para NMake.
CodeView Options:	Controlar como o CodeView utiliza a memória e a informação exibida
Browse Options:	Define a maneira de como o <i>browser</i> é construído.
Fortran Compiler Options:	Escolher as opções de compilação.

## Browse



<b>Goto Definition:</b>	Localizar a definição de qualquer símbolo do programa em seu programa fonte.
<b>Goto Reference:</b>	Localizar a referência para qualquer nome na base de dados do browser.
<b>View Relationship:</b>	Consultar a base de dados do browser.
<b>List References:</b>	Exibir a lista de funções que chama cada função e usa cada variável e tipo.
<b>Call Tree:</b>	Exibir as funções que chamam outras funções.
<b>Outline:</b>	Exibir a saída do programa.
<b>Next:</b>	Encontrar a próxima definição ou referência.
<b>Previous:</b>	Encontrar a definição ou referência anterior.
<b>Case Sensitive:</b>	Definir se a busca no browser é sensítiva.
<b>Split Windows:</b>	Determinar como o browser aparece na tela.

## Help



<b>Index:</b>	Exibir a lista de índices do Fortran (Shift F1).
<b>Contents:</b>	Exibir a tabela de conteúdo dos assuntos explorados pelo Fortran (CTRL F1).
<b>Topic:</b>	Exibir a informação sobre o item ou palavra chave na qual o cursor está posicionado (F1). Por exemplo, posicionando o cursor sobre o comando WRITE que está na janela de edição e escolhendo esta opção (F1) aparecerá uma janela explicando a ação do comando WRITE e a sua sintaxe, evitando, assim, de procurar nos manuais do Fortran o que o comando WRITE faz.
<b>Help on Help:</b>	Explicar como utilizar o sistema de ajuda (help) do ambiente.
<b>Next:</b>	Exibir o próximo pedaço de informação no sistema de ajuda sobre um tópico.

### 3. Edição, Compilação, Composição e Execução de um Programa

Pronto, agora você já conhece as principais opções do menu principal do ambiente Fortran. Vamos praticar! Para isto, siga os passos seguintes.

1. Entrar no ambiente do Fortran.

```
D:\>  
D:\>cd fortran <Enter>  
D:\FORTRAN\>forbat2 <Enter>  
D:\FORTRAN>pw6 <Enter>
```

2. Criar e editar um programa fonte.

- ◆ Escolher a opção **File** do menu principal e depois *New*.
- ◆ Agora digite o programa, utilizando os recursos de edição oferecidos pelo ambiente do Fortran.

O programa COBAIA, mostrado a seguir, será utilizado como exemplo para ilustrar os procedimentos descritos nesta e nas próximas seções.

```
* Este programa calcula e imprime o fatorial de um numero N lido pelo teclado.  
* Autor: Marcelo  
* Data: 22/04/1995  
* Programa Principal  
PROGRAM COBAIA  
* Declaracao das variaveis utilizadas  
INTEGER FAT, M, N  
* Leitura de N  
WRITE(*,*) 'DIGITE O VALOR DE N:'  
READ(*,*) N  
FAT = 1  
DO 10, M=2, N  
    FAT = FAT * M  
10 CONTINUE  
* Imprimir o fatorial  
WRITE(*,20) N, FAT  
20 FORMAT(1X, 'FATORIAL DE ',I2,' = ', I4)  
STOP  
END
```

3. Ao final do processo de edição vamos Salvar o programa com o nome de COBAIA.FOR (o nome COBAIA é arbitrário; somente a extensão FOR é obrigatória).

- ◆ Escolher a opção **File** do menu principal e depois *Save*.

Digite: C:\TMP\COBAIA.FOR <Enter>

4. Neste momento você já pode realizar as etapas de processamento: compilação, composição e execução do programa, como mostradas abaixo.

♦ **Compilação:** Escolher a opção **Make** do menu principal e depois *Compile File*.

No nosso exemplo nenhum erro de compilação será assinalado. Porém, se algum erro for detectado após este primeiro passo, o programa fonte deverá ser retomado com a ajuda do editor, para que as correções possam ser realizadas. Após as correções, o processo de compilação deverá ser reiniciado. Caso não haja erros, passa-se à etapa de composição.

♦ **Composição:** Escolher a opção **Make** do menu principal e depois *Build*.

Alguns erros podem ser detectados nesta etapa; quando isto ocorrer, os erros deverão ser corrigidos no programa fonte e após estas correções, deve ser reiniciado novamente o processo de compilação.

O programa COBAIA.FOR não possui nenhum erro de composição. Assim, vamos executar o programa.

♦ **Execução:** Escolher a opção **Run** do menu principal e depois *Execute*.

Se o programa exigir a entrada de dados (no exemplo COBAIA.FOR é necessário ler o valor de N para o programa calcular o fatorial) eles serão solicitados neste momento. Somente após o fornecimento dos dados de entrada é que serão mostrados os resultados do processamento.

Abaixo mostramos a execução do programa COBAIA.FOR.

```
DIGITE O VALOR DE N:
```

```
_
```

O usuário, por sua vez, digita um valor inteiro.

```
DIGITE O VALOR DE N:
```

```
4 <Enter>
```

O resultado final do programa COBAIA é:

```
FATORIAL DE 4 = 24
```

```
Stop - Program terminated
```

```
Press any key to continue ...
```



Pronto, você já sabe como editar, compilar, compor ("linkedit") e executar um programa escrito em Fortran.

#### 4. Erros

A seqüência ilustrada acima para a qual não houve ocorrência de erros nem sempre reflete a realidade. Usualmente, o usuário comete erros de codificação, de edição ou mesmo erros na solução do problema. A detecção de erros durante as etapas de processamento de um programa é um fato bastante comum.

De forma geral, podemos ter as seguintes categorias de erros:

- ✦ Erros Sintáticos.
- ✦ Erros de Composição.
- ✦ Erros de Execução.
- ✦ Erros Semântico (de Solução).

Nas seções seguintes procuramos ilustrar as situações em que ocorrem esses erros e as sugestões de procedimentos de correção a serem conduzidos pelo programador (aluno).

##### 4.1. Erros Sintáticos

Durante a etapa de compilação, os erros sintáticos são indicados através de números e mensagens de erros resumidas que são mostradas na tela. Mensagens de erros mais detalhadas podem ser encontradas no Guia do Usuário, que acompanha o compilador do Fortran [Microsoft, 1991]. Para facilitar a localização dos erros é apontado, junto com o número do erro, o número da linha do programa fonte na qual provavelmente eles tenham ocorrido. Erros detectados durante a etapa de compilação são, em geral, erros relativos às normas sintáticas da linguagem (Erros Sintáticos). Como já dito, sempre que forem detectados erros na etapa de compilação, o programa fonte deve ser retomado com a ajuda do editor, para que sejam feitas as correções. Após estas correções, deve ser reiniciada a etapa de compilação.

O exemplo apresentado a seguir mostra o programa fonte COBAIA1.FOR onde foram inseridos deliberadamente alguns erros sintáticos, como por exemplo: declaração errônea de variáveis e a falta do número do comando FORMAT.

```

* Este programa calcula e imprime o fatorial de um numero N lido pelo teclado.
* Autor: Marcelo
* Data: 22/04/1995
* Programa Principal
PROGRAM COBAIA1
* Declaracao das variaveis utilizadas
INTOGER FAT, M, N (linha 7)
* Leitura de N
WRITE(*,*) 'DIGITE O VALOR DE N:'
READ(*,*) N
FAT = 1
DO 10, M=2, N
    FAT = FAT * M
10 CONTINUE
* Imprimir o fatorial
WRITE(*,20) N, FAT
FORMAT(1X, 'FATORIAL DE ', I2, ' = ', I4) (linha 17)
STOP
END (linha 19)

```

Ao compilar o programa acima se obtêm os seguintes números e mensagens de erros na janela *Compiler Results*:

```

                                Compiler Results
c:\TMP\COBAIA1.FOR (7): error F2115:      syntaxe error
c:\TMP\COBAIA1.FOR (17): error F2720:     FORMAT: label missing
c:\TMP\COBAIA1.FOR (19): warning F4999:   variable declared but not
used
c:\TMP\COBAIA1.FOR (19): error F2837:     label 20: undefined

```

## 4.2. Erros de Composição

Ainda que não sejam detectados erros na fase de compilação, alguns erros ainda podem surgir durante a execução das outras etapas de processamento. Erros detectados durante a etapa de composição são denominados de *erros de composição* e usualmente se relacionam a referências não resolvidas de funções ou sub-rotinas. Se, por exemplo, ao invés de SIN(X) for escrito SEN(X) em um programa fonte, para indicar a função seno, este erro deverá ser detectado durante a etapa de composição (o editor de ligação não localiza nas bibliotecas a função SEN(X)).

O exemplo apresentado a seguir mostra o programa fonte COBAIA2.FOR onde além de calcular o fatorial de N, calcula o coseno de N. Desta maneira, foram inseridos deliberadamente alguns erros na fase de composição.

```

* Este programa calcula e imprime o fatorial e o coseno de um numero N lido
  pelo teclado.
* Autor: Marcelo
* Data: 22/04/1995
* Programa Principal
PROGRAM COBAIA2
* Declaracao das variaveis utilizadas
INTEGER FAT, M, N
REAL S
* Leitura de N
WRITE(*,*) 'DIGITE O VALOR DE N:'
READ(*,*) N
FAT = 1
DO 10, M=2, N
    FAT = FAT * M
10 CONTINUE
* Imprimir o fatorial
WRITE(*,20) N, FAT
20 FORMAT(1X, 'FATORIAL DE ',I2,' = ', I4)
S = COSENO(N)
WRITE(*,30) N, S
30 FORMAT(1X, 'COSENO DE ',I2,' = ', F4.0)
STOP
END

```

Ao compor o programa acima se obtém o seguinte número e mensagem de erro:

```

                          Compiler Results
C:\TMP\COBAIA2.OBJ (C:\TMP\COBAIA2.FOR):
error L2029:          'COSENO':          unresolved external.

There was 1 error detected.

```

### 4.3. Erros de Execução

Erros também podem ser detectados durante a fase de execução e são denominados *erros de execução*. Eles fazem com que a execução dos programas seja interrompida, ao mesmo tempo em que mostra na tela mensagens indicativas. Estas mensagens nem sempre são explícitas ou fáceis de interpretar. Divisões por zero, raízes quadradas de números negativos, overflow (“estouro” do tamanho de uma variável), erros de compatibilidade de tipos nas especificações de formato, acessos a arquivos não definidos são alguns exemplos comuns de causas de erros de execução.

O exemplo apresentado a seguir mostra o programa fonte COBAIA3.FOR na qual foram inseridos deliberadamente alguns erros de execução. O erro assinalado diz respeito

à incompatibilidade existente no comando FORMAT de número 20 na qual as especificações (F2.0 e F4.0) de formatos para números reais não concordam com o comando WRITE associado, no qual só existem variáveis do tipo inteiro.

```
* Este programa calcula e imprime o fatorial de um numero N lido pelo teclado.
* Autor: Marcelo
* Data: 22/04/1995
* Programa Principal
PROGRAM COBAIA3
* Declaracao das variaveis utilizadas
INTEGER FAT, M, N
* Leitura de N
WRITE(*,*) 'DIGITE O VALOR DE N:'
READ(*,*) N
FAT = 1
DO 10, M=2, N
    FAT = FAT * M
10 CONTINUE
* Imprimir o fatorial
WRITE(*,20) N, FAT
20 FORMAT(1X, 'FATORIAL DE ',F2.0,' = ', F4.0)
STOP
END
```

Ao executar programa acima se obtém o seguinte resultado:

```
DIGITE O VALOR DE N:
4 <Enter>

run-time error F6207: WRITE(CON)
- I edit descriptor expected for Integer.

Press any key to continue ...
```

#### 4.4. Erros Semânticos

Além dos tipos de erros discutidos até agora, outros tipos de erros podem estar associados a um programa Fortran. Este é o caso, por exemplo, dos erros presentes no projeto lógico do programa (algoritmo realizado erroneamente). Esta classe de erros pode provocar resultados indesejáveis de processamento, sem que sejam emitidas mensagens indicativas, em qualquer das etapas deste processamento. Sua correção exige, em geral, uma revisão completa de toda a concepção algorítmica do programa, ao final da qual se devem reiniciar as atividades de processamento. Esses tipos de erros são considerados os erros mais difíceis de serem detectados pelo usuário.

O exemplo apresentado a seguir mostra o programa fonte COBAIA4.FOR onde foi inserido deliberadamente um erro de semântica. O erro semântico ocorre na inicialização da variável FAT (FAT=0 em vez de FAT=1).

```
* Este programa calcula e imprime o fatorial de um numero N lido pelo teclado.
* Autor: Marcelo
* Data: 22/04/1995
* Programa Principal
PROGRAM COBAIA4
* Declaracao das variaveis utilizadas
INTEGER FAT, M, N
* Leitura de N
WRITE(*,*) 'DIGITE O VALOR DE N:'
READ(*,*) N
FAT = 0
DO 10, M=2, N
    FAT = FAT * M
10 CONTINUE
* Imprimir o fatorial
WRITE(*,20) N, FAT
20 FORMAT(1X, 'FATORIAL DE ',I2,' = ', I4)
STOP
END
```

Ao executar programa acima se obtém o resultado:

```
DIGITE O VALOR DE N:
4 <Enter>

FATORIAL DE 4 = 0

Stop - Program terminated

Press any key to continue ...
```

Caso você tenha que escrever o(s) resultado(s) do seu programa na impressora ao invés de exibí-lo(s) na tela do computador é necessário fazer algumas adaptações. Por exemplo, para imprimir o fatorial de um número qualquer na impressora temos que realizar duas modificações no programa COBAIA4:

- 1) Inserir o comando OPEN(6,FILE='PRN').
- 2) No momento que vai imprimir a mensagem colocar no primeiro parâmetro do WRITE o número 6.

O programa COBAIA4 modificado é apresentado a seguir.

## **Bibliografia**

- [Advanced Topics, 1991] *Microsoft FORTRAN — Advanced Topics* (version 5.1). 362p.
- [Cereda & Maldonado, 1987] Cereda, R.L.D. & Maldonado, J.C. *Introdução ao FORTRAN 77 para microcomputadores*. McGraw-Hill, 1987, 211p.
- [Environment and Tools, 1991] *Microsoft FORTRAN — Environment and Tools* (version 5.1). 666p.
- [Microsoft, 1991] *Microsoft FORTRAN — Reference* (version 5.1). 534p.

*Há, verdadeiramente, duas coisas diferentes: saber e crer que se sabe. A ciência consiste em saber; em crer que se sabe reside a ignorância”*

**Hipócrates**