



UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA MECÂNICA

Disciplina: TM-102 INFORMÁTICA / PROGRAMAÇÃO**Semestre: 2003/2**

6ª aula

INSTRUÇÕES

- Siga o roteiro abaixo.
- Qualquer dúvida, chame o professor.

6.1 OBJETIVOS DA AULA

- 1) Ciclos
- 2) Operadores lógicos
- 3) Determinação do tempo de computação

6.2 CRIANDO UM NOVO PROJETO DO TIPO QUICKWIN APPLICATION

- 1) Na “janela principal” do Fortran execute: File, New
- 2) Na janela “New”: selecione “Project Workspace” e clique sobre o botão “OK”
- 3) Na janela “New Project Workspace”, execute o seguinte:
 - a) Type: selecione “QuickWin Application”
 - b) Location: use o botão BROWSE para indicar o seu diretório no “drive” Server1, diretório (pasta) “alunos_2003_2”.
 - c) Name: indique o nome do projeto. Use “programa06”. O software criará um diretório com este nome, dentro do diretório indicado no passo anterior.
 - d) Clique em “Create”
 - e) No lado esquerdo da “janela principal” deverá aparecer o nome do projeto criado

6.3 INSERINDO UM ARQUIVO DENTRO DO PROJETO

- 1) Na “janela principal” do Fortran, execute: Insert, Files into Project
- 2) Na janela “Insert Files into Project”:
 - a) File Name: entre com o nome do arquivo a ser criado. Use “programa6a.f90”. Clique em “OK”
 - b) Na janela “Microsoft Developer Studio”, clique em YES
 - c) No lado esquerdo da “janela principal”, clique sobre o sinal + ao lado do nome do projeto; deverá aparecer o nome do arquivo que foi inserido.
 - d) Clique duas vezes sobre o nome do arquivo que foi inserido
 - e) Na janela “Microsoft Developer Studio”, clique em YES
 - f) O arquivo inserido está pronto para ser editado no lado direito da “janela principal”

6.4 PROGRAMA6a.f90

1) Dentro do espaço de edição de programas do Fortran, digite exatamente o seguinte:

```
use msflib
use msfwin

implicit none

logical dos
logical janela

integer*4 fecha

integer inicio
integer fim
integer passo
integer contador

janela = setwindowtext(gethwn dqq(qwin$framewindow), "PROGRAMA6a.F90 - Ciclos"C)

dos = systemqq('Notepad dados6a.txt')

open(12, file='dados6a.txt')

read(12, *) inicio
read(12, *) fim
read(12, *) passo

close(12)

open(10, file='saida6a.txt')

write(10, 20) inicio, fim, passo

do contador = inicio, fim, passo
    write(10, 30) contador
end do

close(10)

dos = systemqq('Notepad saida6a.txt')

fecha = setexitqq(qwin$exitnopersist)

20 format( 1/, 5x, '***** DADOS *****', &
          1/, 5x, 'inicio = ', i5, &
          1/, 5x, 'fim = ', i5, &
          1/, 5x, 'passo = ', i5, 1/ )

30 format( 3x, 'contador = ', i5 )

end
```

2) Depois de digitar o programa, grave-o da seguinte forma: File, Save

3) Compile o programa executando: Build, Compile

4) Gere o executável do programa com Build, Build

5) Estude os comentários a seguir a respeito de cada linha (em vermelho) do “programa6a.f90”:

```
use msflib
! Inclui no programa a biblioteca do Fortran que permite usar os comandos "systemqq" e
! "setexitqq"
```

use msfwin

! Inclui no programa a biblioteca do Fortran que permite usar o comando "setwindowtext"

implicit none

! Ao se usar este comando, todas as variáveis do programa têm que ser definidas
! explicitamente com os comandos real, character, integer e logical. Assim, evita-se
! erros de lógica no programa, captados ao se fazer a compilação.

logical dos

! Declaração de variável do tipo "lógica" para usar o comando "systemqq"

logical janela

! Declaração de variável do tipo "lógica" para usar o comando "setwindowtext"

integer*4 fecha

! Declaração de variável do tipo "inteiro" para usar o comando "setexitqq"

integer inicio

! Declaração de variável do tipo "inteiro". Ela conterá o valor inicial do "ciclo"

integer fim

! Declaração de variável do tipo "inteiro". Ela conterá o valor final do "ciclo"

integer passo

! Declaração de variável do tipo "inteiro". Ela conterá o passo para alterar a
! variável "contador" do "ciclo"

integer contador

! Declaração de variável do tipo "inteiro". Seu valor irá variar a cada "passo" do
! "ciclo"

janela = setwindowtext(getwndqq(qwin\$framewindow), "PROGRAMA6a.F90 - Ciclos"C)

! Esta instrução define o nome da janela do programa.
! O nome deve ficar entre aspas, conforme o exemplo acima.

dos = systemqq('Notepad dados6a.txt')

! Mostra o conteúdo do arquivo "dados6a.txt" com o programa Notepad

open(12, file='dados6a.txt')

! Define o número "12" como uma referência ao arquivo "dados6a.txt"

read(12,*) inicio

read(12,*) fim

read(12,*) passo

! Utiliza o comando READ para ler as variáveis "inicio", "fim" e "passo" do arquivo 12
! que corresponde ao arquivo "dados6a.txt", conforme definido no comando OPEN acima.
! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer formato.
! Como há um comando READ para cada variável, é lido apenas uma variável por linha no
! arquivo "dados6a.txt".

close(12)

! O comando CLOSE encerra o uso do arquivo 12

open(10, file='saida6a.txt')

! Define o número "10" como uma referência ao arquivo "saida6a.txt"

write(10,20) inicio, fim, passo

! Escreve os valores das variáveis "inicio", "fim" e "passo" no
! arquivo "10" com o formato especificado pelo número "20".

! Um CICLO na linguagem Fortran é definido por uma linha que inicia com o comando DO e
! outra linha que contém o comando END DO.
! Todas as linhas que estiverem entre os comandos DO e END DO serão repetidas de acordo
! com o que se define na linha do comando DO. Esta linha é definida por quatro variáveis:
! (1) um valor inicial do ciclo (neste exemplo, a variável "inicio");
! (2) um valor final do ciclo (neste exemplo, a variável "fim");
! (3) um passo para alterar a variável "contador" do ciclo (neste exemplo, a variável
! "passo"); e
! (4) um contador (neste exemplo, a variável "contador").

! Estas quatro variáveis podem ser do tipo inteiro (como neste exemplo) ou do tipo real.
! O ciclo pode ser progressivo (início < fim) ou regressivo (início > fim).

do contador = inicio, fim, passo

! A variável "contador" começará com o valor da variável "inicio".
! Quando a execução do programa atingir o comando END DO, volta-se ao comando DO e o valor da variável "contador" passará a ser a soma da variável "contador" com "passo".
! Se o novo valor da variável "contador" estiver no intervalo fechado definido pelas variáveis "inicio" e "fim", o programa voltará a executar o ciclo DO a END DO. Caso contrário, a execução do programa prosseguirá a partir do comando END DO.

write(10,30) contador

! Escreve o valor da variável "contador" no arquivo "10" com o formato especificado pelo número "30".

end do

! Encerra o ciclo DO

close(10)

! O comando CLOSE encerra o uso do arquivo 10

dos = systemqq('Notepad saida6a.txt')

! Mostra o conteúdo do arquivo "saida6a.txt" com o programa Notepad

fecha = setexitqq(qwin\$exitnopersist)

! A instrução acima fecha automaticamente a janela do programa ao final de sua execução

```
20 format( 1/, 5x, '***** DADOS *****', &
          1/, 5x, 'inicio = ', i5, &
          1/, 5x, 'fim    = ', i5, &
          1/, 5x, 'passo  = ', i5, 1/ )
```

```
30 format( 3x, 'contador = ', i5 )
```

end

! O comando END encerra o programa

6) Antes de executar este programa, você precisa criar o arquivo “dados6a.txt”, inserindo os três dados que são necessários para o programa6a.f90. Para fazer isso, execute o seguinte:

a) Start, Programs, Accessories, Notepad

b) Dentro do espaço de edição do Notepad, edite por exemplo:

```
10      inicio
20      fim
1       passo
```

c) File, Save

d) Na janela “Save As”, em “File Name”, coloque “dados6a.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.

7) Execute o programa com Build, Execute. O resultado deverá ser:

a) A abertura de uma janela do DOS.

b) A abertura de uma janela do WINDOWS que terá como título “Programa6a.f90 - CICLOS”, conforme foi especificado dentro do programa6a.f90.

c) A abertura do arquivo “dados6a.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se você quiser, altere o conteúdo das variáveis no arquivo “dados6a.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.

d) Em seguida, ocorrerá a abertura do arquivo “saida6a.txt” com o programa Notepad. O conteúdo deverá ser:

```
***** DADOS *****
inicio = 10
fim    = 20
passo  = 1

contador = 10
```

```
contador = 11
contador = 12
contador = 13
contador = 14
contador = 15
contador = 16
contador = 17
contador = 18
contador = 19
contador = 20
```

- e) Até entender, compare o conteúdo de cada linha do arquivo “saida6a.txt” com os comandos WRITE do programa6a.f90 e seus respectivos formatos de edição.
- f) Feche o arquivo “saida6a.txt” com File, Exit.
- g) Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.
- 8) Execute novamente o programa com Build, Execute. Altere os dados para: 10, 20 e 2. Analise os resultados.
- 9) Execute novamente o programa com Build, Execute. Altere os dados para: 10, 20 e 3. Analise os resultados.
- 10) Execute novamente o programa com Build, Execute. Altere os dados para: 10, 0 e -2. Analise os resultados.
- 11) Execute novamente o programa com Build, Execute. Altere os dados para: -5, 5 e 1. Analise os resultados.

6.5 PROGRAMA6b.f90

- 1) Clique dentro do espaço de edição do seu “programa6a.f90”. Depois execute: File, Close. Clique sobre o nome do “programa6a.f90”, no lado esquerda da janela principal do Fortran. Em seguida, execute: Edit, Cut. Com isso, o “programa6a.f90” será eliminado do seu projeto, mas o arquivo continuará a existir dentro do diretório do seu projeto, podendo ser incluído nele quando desejado.
- 2) Seguindo os passos da seção 6.3 deste tutorial, insira no projeto “programa06” o arquivo “programa6b.f90”.
- 3) Dentro do espaço de edição de programas do Fortran, digite exatamente o seguinte:

```
use msflib
use msfwin

implicit none

logical dos

logical janela

integer*4 fecha

integer inicio
integer fim
integer passo
integer contador
integer soma_positivos
integer soma_negativos

janela = setwindowtext(gethwndq(qwin$framewindow), &
    "Programa6b.f90 - OPERADORES LÓGICOS"C)

dos = systemqq('Notepad dados6b.txt')

open(12,file='dados6b.txt')

read(12,*) inicio
read(12,*) fim
read(12,*) passo
```

```
close(12)

open(10,file='saida6b.txt')

write(10,20) inicio, fim, passo

if ( inicio < fim ) write(10,40)
if ( inicio <= fim ) write(10,41)
if ( inicio > fim ) write(10,50)
if ( inicio >= fim ) write(10,51)
if ( inicio == fim ) write(10,60)
if ( inicio /= fim ) write(10,61)

if ( inicio < 0 .and. fim < 0 ) write(10,120)
if ( inicio > 0 .and. fim > 0 ) write(10,130)
if ( inicio > 0 .or. fim > 0 ) write(10,140)

write(10,*)

soma_positivos = 0
soma_negativos = 0

do contador = inicio, fim, passo

    write(10,30) contador

    if ( contador > 0 ) then
        soma_positivos = soma_positivos + contador
        if ( soma_positivos > fim ) then
            write(10,300)
            exit
        end if
    else
        soma_negativos = soma_negativos + contador
    end if

end do

write(10,200) soma_negativos, soma_positivos

close(10)

dos = systemqq('Notepad saida6b.txt')

fecha = setexitqq(qwin$exitnopersist)

20 format( 1/, 5x, '***** DADOS *****', &
          1/, 5x, 'inicio = ', i5, &
          1/, 5x, 'fim    = ', i5, &
          1/, 5x, 'passo  = ', i5, 1/ )

30 format( 3x, 'contador = ', i5 )

40 format( '"inicio" é MENOR DO QUE "fim"' )

41 format( '"inicio" é MENOR OU IGUAL A "fim"' )

50 format( '"inicio" é MAIOR DO QUE "fim"' )

51 format( '"inicio" é MAIOR OU IGUAL A "fim"' )
```

```
60 format( '"inicio" é IGUAL A "fim"' )
61 format( '"inicio" é DIFERENTE DE "fim"' )
120 format( '"inicio" E "fim" são números negativos' )
130 format( '"inicio" E "fim" são números positivos' )
140 format( '"inicio" OU "fim" é um número positivo' )
200 format( 1/, 5x, 'soma de negativos = ', i5, &
           2/, 5x, 'soma de positivos = ', i5 )
300 format( 1/, 'executado o comando EXIT', &
           ' devido "soma_positivos" ser MAIOR que "fim"' )

end
```

- 4) Depois de digitar o programa, grave-o da seguinte forma: File, Save
- 5) Compile o programa executando: Build, Compile
- 6) Gere o executável do programa com Build, Build
- 7) Estude os comentários a seguir a respeito de cada linha (em vermelho) do “programa6b.f90”. Só foram incluídos as linhas e os comentários do “programa6b.f90” que são adicionais às linhas do “programa6a.f90”.

```
integer soma_positivos
! Declaração de variável do tipo "inteiro". Ela conterà a soma dos números positivos

integer soma_negativos
! Declaração de variável do tipo "inteiro". Ela conterà a soma dos números negativos

! O comando IF (SE) é usado para fazer testes lógicos.

! Neste programa são exemplificadas duas formas do comando IF.

! Na primeira forma do comando IF, usa-se:
! IF ( condição ) comando
! Nesta forma, uma condição lógica deve ser colocada dentro de parênteses.
! Se a condição for satisfeita, o "comando" fora do parênteses é executado.
! Se a condição não for satisfeita, o programa executa a linha seguinte ao comando IF.
! Alguns exemplos são apresentados a seguir.

if ( inicio < fim ) write(10,40)
! SE a variável "inicio" for MENOR (<) DO QUE a variável "fim",
! o comando WRITE(10,40) é executado; caso contrário, não.

if ( inicio <= fim ) write(10,41)
! SE a variável "inicio" for MENOR OU IGUAL (<=) à variável "fim",
! o comando WRITE(10,41) é executado; caso contrário, não.

if ( inicio > fim ) write(10,50)
! SE a variável "inicio" for MAIOR (>) DO QUE a variável "fim",
! o comando WRITE(10,50) é executado; caso contrário, não.

if ( inicio >= fim ) write(10,51)
! SE a variável "inicio" for MAIOR OU IGUAL (>=) à variável "fim",
! o comando WRITE(10,51) é executado; caso contrário, não.

if ( inicio == fim ) write(10,60)
! SE a variável "inicio" for IGUAL (==) à variável "fim",
! o comando WRITE(10,60) é executado; caso contrário, não.

if ( inicio /= fim ) write(10,61)
! SE a variável "inicio" for DIFERENTE (/=) da variável "fim",
! o comando WRITE(10,61) é executado; caso contrário, não.
```

! A condição do comando IF, na verdade, pode ser composta por mais de uma condição
! conforme os exemplos a seguir. Nestes casos, as várias condições devem ser unidas
! pelos OPERADORES LÓGICOS "AND" ou "OR" que, em português, significam "E" ou "OU".

```
if ( inicio < 0 .and. fim < 0 ) write(10,120)
```

! SE a variável "inicio" for MENOR (<) DO QUE o número zero (0) E
! SE a variável "fim" for MENOR (<) DO QUE o número zero (0)
! o comando WRITE(10,120) é executado; caso contrário, não.

```
if ( inicio > 0 .and. fim > 0 ) write(10,130)
```

! SE a variável "inicio" for MAIOR (>) DO QUE o número zero (0) E
! SE a variável "fim" for MAIOR (>) DO QUE o número zero (0)
! o comando WRITE(10,130) é executado; caso contrário, não.

```
if ( inicio > 0 .or. fim > 0 ) write(10,140)
```

! SE a variável "inicio" for MAIOR (>) DO QUE o número zero (0) OU
! SE a variável "fim" for MAIOR (>) DO QUE o número zero (0)
! o comando WRITE(10,140) é executado; caso contrário, não.

```
write(10,*)
```

```
soma_positivos = 0
```

! Atribui (fornece, define) o número zero (0) à variável "soma_positivos"

```
soma_negativos = 0
```

! Atribui (fornece, define) o número zero (0) à variável "soma_negativos"

! Na segunda forma do comando IF, usa-se:

```
! IF ( condição ) THEN
!   comandos (uma ou várias linhas de programa)
! ELSE
!   comandos (uma ou várias linhas de programa)
! END IF
```

! Nesta forma, uma condição lógica também deve ser colocada dentro de parênteses.

! Se a condição for satisfeita, os comandos que estão entre o IF e o ELSE serão executados.

! Se a condição não for satisfeita, o programa executará os comandos que estão entre o ELSE e o END IF.

! Pode-se usar misturas das duas formas do comando IF descritas acima.

! Alguns exemplos são apresentados a seguir dentro de um ciclo DO a END DO.

```
do contador = inicio, fim, passo
```

! inicializa um CICLO

```
write(10,30) contador
```

```
if ( contador > 0 ) then
```

```
  soma_positivos = soma_positivos + contador
```

```
  if ( soma_positivos > fim ) then
```

```
    write(10,300)
```

```
    exit
```

! o comando EXIT é usado para que o programa encerre a execução de um ciclo DO,
! passando a executar a linha seguinte ao comando END DO.

```
  end if
```

```
else
```

```
  soma_negativos = soma_negativos + contador
```

```
end if
```

```
end do
```

8) Antes de executar este programa, você precisa criar o arquivo "dados6b.txt", inserindo os três dados que são necessários para o programa6b.f90. Para fazer isso, execute o seguinte:

a) Start, Programs, Accessories, Notepad

b) Dentro do espaço de edição do Notepad, edite por exemplo:

```
10      inicio
```

```
20      fim
```


1 passo

- c) File, Save
- d) Na janela “Save As”, em “File Name”, coloque “dados6b.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.
- 9) Execute o programa com Build, Execute. O resultado deverá ser:
- A abertura de uma janela do DOS.
 - A abertura de uma janela do WINDOWS que terá como título “Programa6b.f90 - OPERADORES LÓGICOS”, conforme foi especificado dentro do programa6b.f90.
 - A abertura do arquivo “dados6b.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se você quiser, altere o conteúdo das variáveis no arquivo “dados6b.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.
 - Em seguida, ocorrerá a abertura do arquivo “saida6b.txt” com o programa Notepad. O conteúdo deverá ser:

```
***** DADOS *****
inicio = 10
fim = 20
passo = 1
```

```
"inicio" é MENOR DO QUE "fim"
"inicio" é MENOR OU IGUAL A "fim"
"inicio" é DIFERENTE DE "fim"
"inicio" E "fim" são números positivos
"inicio" OU "fim" é um número positivo
```

```
contador = 10
contador = 11
```

```
executado o comando EXIT devido "soma_positivos" ser MAIOR que "fim"
```

```
soma de negativos = 0
```

```
soma de positivos = 21
```

- Até entender, compare o conteúdo de cada linha do arquivo “saida6b.txt” com a lógica e os comandos WRITE do programa6b.f90 e seus respectivos formatos de edição.
 - Feche o arquivo “saida6b.txt” com File, Exit.
 - Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.
- 10) Execute novamente o programa com Build, Execute. Altere os dados para: 10, 0 e -2. Analise os resultados.
- 11) Execute novamente o programa com Build, Execute. Altere os dados para: 10, 10 e 1. Analise os resultados.
- 12) Execute novamente o programa com Build, Execute. Altere os dados para: -5, 5 e 1. Analise os resultados.
- 13) Execute novamente o programa com Build, Execute. Altere os dados para: -10, -1 e 1. Analise os resultados.

6.6 PROGRAMA6c.f90

- Clique dentro do espaço de edição do seu “programa6b.f90”. Depois execute: File, Close. Clique sobre o nome do “programa6b.f90”, no lado esquerda da janela principal do Fortran. Em seguida, execute: Edit, Cut. Com isso, o “programa6b.f90” será eliminado do seu projeto, mas o arquivo continuará a existir dentro do diretório do seu projeto, podendo ser incluído nele quando desejado.
- Seguindo os passos da seção 6.3 deste tutorial, insira no projeto “programa06” o arquivo “programa6c.f90”.
- Dentro do espaço de edição de programas do Fortran, digite exatamente o seguinte:

```
use msflib
use msfwin
use portlib
```

```
implicit none
```

```
logical dos
logical janela

integer*4 fecha

real*8 inicio
real*8 fim
real*8 passo
real*8 contador
real*8 soma

integer repete

real*8 tempo_cpu

janela = setwindowtext(gethwndqq(qwin$framewindow), &
    "Programa6c.f90 - TEMPO DE PROCESSAMENTO DO COMPUTADOR"C)

dos = systemqq('Notepad dados6c.txt')

open(12,file='dados6c.txt')

read(12,*) inicio
read(12,*) fim
read(12,*) passo

close(12)

open(10,file='saida6c.txt')

write(10,20) inicio, fim, passo

repete = (fim - inicio) / passo

soma = 0.0

tempo_cpu = TIMEF()

do contador = inicio, fim, passo

    soma = soma + contador

end do

tempo_cpu = TIMEF()

write(10,30) soma, tempo_cpu, repete

close(10)

dos = systemqq('Notepad saida6c.txt')

fecha = setexitqq(qwin$exitnopersist)

20 format( 1/, 5x, '***** DADOS *****', &
    1/, 5x, 'inicio = ', 1pe15.5, &
    1/, 5x, 'fim = ', 1pe15.5, &
    1/, 5x, 'passo = ', 1pe15.5 )

30 format( 1/, 1x, 'soma = ', 1pe15.5, &
    2/, 1x, 'tempo de processamento (segundos) = ', 1pe15.5, &
    2/, 1x, 'número de vezes que o ciclo foi realizado = ', i12 )

end
```

- 4) Depois de digitar o programa, grave-o da seguinte forma: File, Save
- 5) Compile o programa executando: Build, Compile
- 6) Gere o executável do programa com Build, Build
- 7) Estude os comentários a seguir a respeito de cada linha (em vermelho) do “programa6c.f90”.

use msflib

! Inclui no programa a biblioteca do Fortran que permite usar os comandos "systemqq" e "setexitqq"

use msfwin

! Inclui no programa a biblioteca do Fortran que permite usar o comando "setwindowtext"

use portlib

! Inclui no programa a biblioteca do Fortran que permite usar o comando "TIMEF"

implicit none

! Ao se usar este comando, todas as variáveis do programa têm que ser definidas explicitamente com os comandos real, character, integer e logical. Assim, evita-se erros de lógica no programa, captados ao se fazer a compilação.

logical dos

! Declaração de variável do tipo "lógica" para usar o comando "systemqq"

logical janela

! Declaração de variável do tipo "lógica" para usar o comando "setwindowtext"

integer*4 fecha

! Declaração de variável do tipo "inteiro" para usar o comando "setexitqq"

real*8 inicio

! Declaração de variável do tipo "real dupla". Ela conterá o valor inicial do "ciclo"

real*8 fim

! Declaração de variável do tipo "real dupla". Ela conterá o valor final do "ciclo"

real*8 passo

! Declaração de variável do tipo "real dupla". Ela conterá o passo para alterar a variável "contador" do "ciclo"

real*8 contador

! Declaração de variável do tipo "real dupla".
! Seu valor irá variar a cada "passo" do "ciclo"

real*8 soma

! Declaração de variável do tipo "real dupla". Ela conterá a soma de todos os valores assumidos pela variável "contador".

integer repete

! Declaração de variável do tipo "inteiro". Ela conterá um valor que representa o número de vezes que o ciclo DO será realizado.

real*8 tempo_cpu

! Declaração de variável do tipo "real dupla". Ela conterá o tempo de processamento do computador para executar o "ciclo" DO.

**janela = setwindowtext(gethwndqq(qwin\$framewindow), &
"Programa6c.f90 - TEMPO DE PROCESSAMENTO DO COMPUTADOR"C)**

! Esta instrução define o nome da janela do programa.
! O nome deve ficar entre aspas, conforme o exemplo acima.

dos = systemqq('Notepad dados6c.txt')

! Mostra o conteúdo do arquivo "dados6c.txt" com o programa Notepad

open(12, file='dados6c.txt')

! Define o número "12" como uma referência ao arquivo "dados6c.txt"

```
read(12,*) inicio
```

```
read(12,*) fim
```

```
read(12,*) passo
```

```
! Utiliza o comando READ para ler as variáveis "inicio", "fim" e "passo" do arquivo 12  
! que corresponde ao arquivo "dados6c.txt", conforme definido no comando OPEN acima.  
! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer formato.  
! Como há um comando READ para cada variável, é lido apenas uma variável por linha no  
! arquivo "dados6a.txt".
```

```
close(12)
```

```
! O comando CLOSE encerra o uso do arquivo 12
```

```
open(10,file='saida6c.txt')
```

```
! Define o número "10" como uma referência ao arquivo "saida6a.txt"
```

```
write(10,20) inicio, fim, passo
```

```
! Escreve os valores das variáveis "inicio", "fim" e "passo" no  
! arquivo "10" com o formato especificado pelo número "20".
```

```
repete = (fim - inicio) / passo
```

```
! Cálculo do número de vezes que o ciclo DO será realizado.
```

```
soma = 0.0
```

```
! Atribui (fornece, define) o número zero (0) à variável "soma"
```

```
tempo_cpu = TIMEF()
```

```
! Dentro de um programa, a primeira chamada do comando TIMEF zera a contagem de tempo.  
! Portanto, a variável "tempo_cpu" será inicializada com o valor zero.
```

```
! Um CICLO na linguagem Fortran é definido por uma linha que inicia com o comando DO e  
! outra linha que contém o comando END DO.
```

```
! Todas as linhas que estiverem entre os comandos DO e END DO serão repetidas de acordo  
! com o que se define na linha do comando DO. Esta linha é definida por quatro variáveis:  
! (1) um valor inicial do ciclo (neste exemplo, a variável "inicio");  
! (2) um valor final do ciclo (neste exemplo, a variável "fim");  
! (3) um passo para alterar a variável "contador" do ciclo (neste exemplo, a variável  
! "passo"); e  
! (4) um contador (neste exemplo, a variável "contador").  
! Estas quatro variáveis podem ser do tipo inteiro (como neste exemplo) ou do tipo real.  
! O ciclo pode ser progressivo (inicio < fim) ou regressivo (inicio > fim).
```

```
do contador = inicio, fim, passo
```

```
! A variável "contador" começará com o valor da variável "inicio".  
! Quando a execução do programa atingir o comando END DO, volta-se ao comando DO e o  
! valor da variável "contador" passará a ser a soma da variável "contador" com "passo".  
! Se o novo valor da variável "contador" estiver no intervalo fechado definido pelas  
! variáveis "inicio" e "fim", o programa voltará a executar o ciclo DO a END DO. Caso  
! contrário, a execução do programa prosseguirá a partir do comando END DO.
```

```
    soma = soma + contador
```

```
    ! adiciona os valores atuais das variáveis "soma" e "contador"
```

```
    ! e atribui o resultado à variável "soma"
```

```
end do
```

```
! Encerra o ciclo DO
```

```
tempo_cpu = TIMEF()
```

```
! Dentro de um programa, a segunda chamada do comando TIMEF contém o tempo de  
! processamento do computador desde a primeira chamada do comando TIMEF.  
! Portanto, a variável "tempo_cpu" terá um valor que corresponde ao  
! tempo de processamento do computador, em segundos.
```

```
write(10,30) soma, tempo_cpu, repete
```

```
! Escreve os valores das variáveis "soma", "tempo_cpu" e "repete" no  
! arquivo "10" com o formato especificado pelo número "30".
```

```
close(10)
```

```
! O comando CLOSE encerra o uso do arquivo 10
```

```
dos = systemqq('Notepad saida6c.txt')
```

! Mostra o conteúdo do arquivo "saida6c.txt" com o programa Notepad

```
fecha = setexitqq(qwin$exitnopersist)
```

! A instrução acima fecha automaticamente a janela do programa ao final de sua execução

```
! *** bloco dos formatos ***
```

```
20 format( 1/, 5x, '***** DADOS *****', &
          1/, 5x, 'inicio = ', 1pe15.5, &
          1/, 5x, 'fim    = ', 1pe15.5, &
          1/, 5x, 'passo  = ', 1pe15.5 )
```

```
30 format( 1/, 1x, 'soma = ', 1pe15.5, &
          2/, 1x, 'tempo de processamento (segundos) = ', 1pe15.5, &
          2/, 1x, 'número de vezes que o ciclo foi realizado = ', i12 )
```

```
end
```

! O comando END encerra o programa

8) Antes de executar este programa, você precisa criar o arquivo “dados6c.txt”, inserindo os três dados que são necessários para o programa6b.f90. Para fazer isso, execute o seguinte:

a) Start, Programs, Accessories, Notepad

b) Dentro do espaço de edição do Notepad, edite por exemplo:

```
1.0      inicio
2.0      fim
1.0e-8   passo
```

c) File, Save

d) Na janela “Save As”, em “File Name”, coloque “dados6c.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.

9) Execute o programa com Build, Execute. O resultado deverá ser:

a) A abertura de uma janela do DOS.

b) A abertura de uma janela do WINDOWS que terá como título “Programa6c.f90 - TEMPO DE PROCESSAMENTO DO COMPUTADOR”, conforme foi especificado dentro do programa6c.f90.

c) A abertura do arquivo “dados6c.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se você quiser, altere o conteúdo das variáveis no arquivo “dados6c.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.

d) Em seguida, ocorrerá a abertura do arquivo “saida6c.txt” com o programa Notepad. O conteúdo deverá ser:

```
***** DADOS *****
```

```
inicio =      1.00000E+00
```

```
fim      =      2.00000E+00
```

```
passo    =      1.00000E-08
```

```
soma =      1.50000E+08
```

```
tempo de processamento (segundos) =      1.08200E+00
```

```
número de vezes que o ciclo foi realizado =      10000000
```

e) Este tempo de processamento refere-se a um microcomputador Pentium III de 750 MHz. Provavelmente o seu tempo de processamento será diferente pois ele depende do tipo de processador usado em cada microcomputador.

f) Até entender, compare o conteúdo de cada linha do arquivo “saida6c.txt” com a lógica e os comandos WRITE do programa6c.f90 e seus respectivos formatos de edição.

g) Feche o arquivo “saida6c.txt” com File, Exit.

h) Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.

10) Execute novamente o programa com Build, Execute. Altere apenas o valor do passo para 1.0e-7 e veja os novos resultados.

6.7 FECHANDO UM PROJETO

- 1) Dentro da “janela principal” do Fortran, feche o seu projeto que está aberto. Para fazer isso, basta executar o seguinte: File, Close Workspace
- 2) Na janela “Microsoft Developer Studio” clique sobre o botão “Yes”.

6.8 ENCERRANDO A AULA

- 1) Dentro da “janela principal” do Fortran, feche o seu projeto executando o seguinte: File, Close Workspace
- 2) Na janela “Microsoft Developer Studio” clique sobre o botão “Yes”.
- 3) Para fechar o programa Fortran, basta executar: File, Exit
- 4) Para fechar o Windows, execute: Start, Shut Down
- 5) Na janela “Shut Down Windows”, escolha a opção “Close all programs and log on as a different user?”. Depois clique no botão “Yes”.
- 6) Depois que o computador sair do Windows, por favor, apenas desligue o monitor do computador.

LISTA DE EXERCÍCIOS DA 6ª AULA

(Para ser entregue na próxima aula)

Exercício 6.1

- 1) Implemente um programa em Fortran para calcular o fatorial de um número N, considerando:
 - a) Ler o número N (inteiro) do arquivo “dados6.txt”.
 - b) Calcular o resultado: o valor (inteiro) do fatorial de N.
 - c) Escrever no arquivo “saida6.txt” o dado lido do arquivo “dados6.txt” e o resultado (fatorial de N), incluindo comentários para informar ao que corresponde cada valor.
- 2) Compile o programa
- 3) Gere o executável do programa
- 4) Execute o programa
- 5) Entregue na próxima aula uma impressão do seu programa fonte em Fortran, do arquivo “dados6.txt” e do arquivo “saida6.txt”

Exercício 6.2

- 1) Acesse o site da disciplina no endereço:
ftp://ftp.demec.ufpr.br/Disciplinas/Tm102/Marchi/semestre_2003_2/
- 2) Estude a página 23 do capítulo 6 da “apostila_logica_programacao.pdf”.
- 3) Estude as páginas 26 a 28 do capítulo 7 da “apostila_logica_programacao.pdf”.