

Computação e Programação

Jaime Ramos e Francisco Miguel Dionísio

Introdução ao Fortran

1 Introdução

Esta aula prática tem como objectivo a apresentação e familiarização com a criação, edição e compilação de programas *Fortran* em ambiente *Eclipse*.

2 Primeiro programa

Inicie a aplicação *Eclipse* e aceite a área de trabalho (*workspace*) sugerida pela aplicação. O primeiro passo a dar é a criação de um novo projecto:

1. No menu *File* seleccione a opção *New* e em seguida a opção *Project*. Na janela que aparece, escolha a opção *Fortran Project* e carregue em *Next*;
2. Escolha um nome para o projecto e nas opções abaixo certifique-se que escolhe *Empty Project - Fortran* na opção *Makefile project*. Carregue em **Next** e não em **Finish**.
3. Na janela seguinte, carregue na etiqueta *Advanced Settings* e escolha a opção¹ **GNU Elf parser**, dentro da opção *Fortran Build - Settings*, na janela do lado esquerdo. Carregue na etiqueta *Error parsers* e escolha a opção **Photran error parser for G95 Fortran**.
4. Carregue em *Ok* e depois em *Finish*.

O próximo passo é escrever o programa:

1. No menu *File* seleccione a opção *New* e em seguida a opção *Source file*. Escolha o nome **exemplo.f95** para o ficheiro, e carregue em *Finish*.
2. Na janela de edição de programas insira o seguinte programa:

```
program exemplo  
  
print *, "1+1=", 1+1  
  
end program exemplo
```

3. Grave o ficheiro.

Para compilar o programa, é necessário criar um novo ficheiro, com a lista de comandos a executar. Este ficheiro chama-se **Makefile**.

1. No menu *File* seleccione a opção *New* e em seguida a opção *File*. Escolha o nome **Makefile** para o ficheiro, e carregue em *Finish*.
2. Na janela de edição de comandos escreva:

¹Esta opção destina-se a compilar programas em ambiente Linux. Noutros ambientes devem usar-se os *parsers* adequados.

```
all:
    g95 -std=F exemplo.f95 -o exemplo.out
```

Note que o espaço no início da segunda linha (antes de `g95`) é obrigatório e é feito com um *tab*.

3. Grave o ficheiro.
4. No menu *Project* desactive a opção *Build automatically*.
5. Para compilar, escolha a opção *Build project* no menu *Project*.
6. Para executar o programa, caso não tenham ocorrido erros, carregue com o botão direito do rato sobre o ficheiro `exemplo.out` que aparece na janela do lado esquerdo e escolha *Run As* e depois *Run Local Fortran Application*.
7. Se aparecer uma janela de diálogo, carregue em *Ok*.
8. O resultado da execução do programa aparece na janela *Console* na parte da baixo do ecrã.

3 Outros exemplo

1. Recorde a função `prime` definida nas aulas teóricas. Escreva o programa seguinte para testar essa função, que solicita ao utilizador um número inteiro e diz se esse número é ou não um número primo:

```
program primo

integer :: n

print *, "Introduza um numero:"
read *, n

if (prime(n)) then
    print *, "O numero e primo."
else
    print *, "O numero nao e primo."
end if

contains

function prime(n) result(b)
integer, intent(in) :: n
logical :: b
integer :: i, d

d=0
do i=1,n
    if (modulo(n,i)==0) then
        d=d+1
    end if
end do
if (d==2) then
    b=.true.
else
    b=.false.
end if
end function prime
```

```
end program primo
```

2. Defina a função `conta_primos` que recebe um número inteiro e devolve o número de números primos que existem até `n`. Experimente a função (sugestão: adapte o programa anterior).
3. Defina a função `ocorrencias` que dado um número `n` e um vector `v` devolve o número de ocorrências de `n` em `v`.
4. Defina a função `car_primos` que dado um vector `v` devolve um vector de igual comprimento com 1 nas posições onde `v` contém um número primo e 0 nas restantes.
5. Defina, recursivamente, a função `pertenceR` que dado um número `n` e um vector `v` devolve `.true.` se `n` ocorre em `v`, e devolve `.false.` caso contrário.