

CAP-372 / 2019

Primeira Lista de Exercícios

Aluno: xxxxxxxx xxxxxxxx xxxxxxxx

Data: 13/08/19

xxxxxxx.xxxxxxx@login.sdumont.lncc.br's password:



Manual: http://sdumont.lncc.br/support_manual.php

The available softwares can be listed with the command: `module avail`
If there's something missing, please get in contact with helpdesk-
sdumont@lncc.br

Respostas dos exercícios da Lista 1:

Objetivo: permitir uma maior familiarização com o ambiente do nó de acesso do sistema Santos Dumont do LNCC

1)

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ pwd
/prj/padinpe/xxxxxxx.xxxxxxx
[xxxxxxx.xxxxxxx@sdumont14 ~]$ whoami
xxxxxxx.xxxxxxx
[xxxxxxx.xxxxxxx@sdumont14 ~]$ ls -la
total 352
drwxr-x---. 2 xxxxxx.xxxxxxx padinpe 158 Jul 24 14:44 .
drwxrwx---. 19 root padinpe 1310 Ago 5 10:51 ..
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 18 Jul 24 14:44 .bash_logout
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 193 Jul 24 14:44 .bash_profile
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 231 Jul 24 14:44 .bashrc
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 334 Jul 24 14:44 .emacs
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 172 Jul 24 14:44 .kshrc
-rw-r--r--. 1 xxxxxx.xxxxxxx padinpe 658 Jul 24 14:44 .zshrc
[xxxxxxx.xxxxxxx@sdumont14 ~]$ gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ module avail
```

```
----- /usr/share/Modules/modulefiles  
-----
```

abyss/2.1	metis/5.1.0_gnu
align_m/2.3	metis/5.1.0_intel
anaconda2/2018.12	modelgenerator/85
anaconda3/2018.12	module-git
bcftools/1.9	module-info
beagle/current	modules
beast/1.10	mummer/4.0
beast/2.5	mumps/5.1.2_intel19

(mostrando somente as 10 primeiras linhas)

2)

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ module load intel_psxe/2019
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ module load gcc/6.5
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ module load gcc/8.3
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ icc -V
```

```
Intel(R) C Intel(R) 64 Compiler for applications running on Intel(R) 64, Version  
19.0.3.199 Build 20190206
```

```
Copyright (C) 1985-2019 Intel Corporation. All rights reserved.
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ ifort -V
```

```
Intel(R) Fortran Intel(R) 64 Compiler for applications running on Intel(R) 64,  
Version 19.0.3.199 Build 20190206
```

```
Copyright (C) 1985-2019 Intel Corporation. All rights reserved.
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ gcc -v
```

```
Using built-in specs.
```

```
COLLECT_GCC=gcc
```

```
COLLECT_LTO_WRAPPER=/scratch/app/gcc/8.3/libexec/gcc/x86_64-pc-linux-gnu/8.3.0/  
lto-wrapper
```

```
Target: x86_64-pc-linux-gnu
```

```
Configured with: ../configure --prefix=/scratch/app/gcc/8.3 --enable-threads --  
enable-lto --disable-multilib
```

```
Thread model: posix
```

```
gcc version 8.3.0 (GCC)
```

```
[xxxxxxx.xxxxxxx@sdumont14 ~]$ gfortran -v
```

```
Using built-in specs.
```

```
COLLECT_GCC=gfortran
```

```
COLLECT_LTO_WRAPPER=/scratch/app/gcc/8.3/libexec/gcc/x86_64-pc-linux-gnu/8.3.0/  
lto-wrapper
```

```
Target: x86_64-pc-linux-gnu
```

```
Configured with: ../configure --prefix=/scratch/app/gcc/8.3 --enable-threads --  
enable-lto --disable-multilib
```

```
Thread model: posix
```

```
gcc version 8.3.0 (GCC)
```

Respostas dos exercícios da Lista 2:

Objetivo: avaliar o uso do comando KIND

“The KIND of a variable is an integer label which tells the compiler which of its supported kinds it should use.”

<https://stackoverflow.com/questions/838310/fortran-90-kind-parameter>

1)

GFORTTRAN

```
[xxxxxxxxxxxxxxxx@sumont14 ~]$ gfortran lista2.f90
```

```
[xxxxxxxxxxxxxxxx@sumont14 ~]$ ./a.out
```

```
DOUBLE PRECISION A8 = 5.5_8      5.5000000000000000
DOUBLE PRECISION A8 = 5.5D00    5.5000000000000000
REAL A4 = 5.5_8      5.50000000
```

```
REAL (KIND = 8) AJ; KIND(AJ) =      8
```

```
REAL (KIND = KIND(0.0D00)) AK; KIND(AK) =      8
```

```
KIND(5.5D00) =      8
```

```
A8 = 5.123456789123456789_8; PRINT A8      5.1234567891234564
A8 = 5.123456789123456789D00; PRINT A8    5.1234567891234564
A4 = 5.123456789123456789D00; PRINT A4    5.12345695
```

```
KIND(I2) - I2 tem quantos bytes?      2
KIND(I4) - I4 tem quantos bytes?      4
KIND(I8) - I8 tem quantos bytes?      8
```

```
KIND(A4) - A4 tem quantos bytes?      4
KIND(A8) - A8 tem quantos bytes?      8
KIND(A16) - A16 tem quantos bytes?    16
```

```
PRECISION(A4) - A4 tem quantos algarismos na mantissa?      6
PRECISION(A8) - A8 tem quantos algarismos na mantissa?    15
PRECISION(A16) - A16 tem quantos algarismos na mantissa?   33
```

```
RANGE(A4) - A4 máximo expoente decimal?      37
RANGE(A8) - A8 máximo expoente decimal?    307
RANGE(A16) - A16 máximo expoente decimal?   4931
```

```
Quero mantissa 20 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)      16
Quero mantissa 10 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)      8
Quero mantissa 05 alg. e exp.dec.020 - SELECTED_REAL_KIND(20,100)      4
```

```
Quero inteiro exp.dec.100 SELECTED_INT_KIND(100)      -1
Quero inteiro exp.dec.10 SELECTED_INT_KIND(10)      8
Quero inteiro exp.dec.09 SELECTED_INT_KIND(09)      4
Quero inteiro exp.dec.04 SELECTED_INT_KIND(09)      2
```

```
I2 = 32000; PRINT I2      32000
I2 = 33000; PRINT I2 - ERROR !!!
```

INTEL

```
[xxxxxxx.xxxxxxx@sumont14 ~]$ ifort lista2.f90  
[xxxxxxx.xxxxxxx@sumont14 ~]$ ./a.out
```

```
DOUBLE PRECISION A8 = 5.5_8      5.5000000000000000  
DOUBLE PRECISION A8 = 5.5D00    5.5000000000000000  
REAL A4 = 5.5_8      5.500000  
  
REAL (KIND = 8) AJ; KIND(AJ) =      8  
  
REAL (KIND = KIND(0.0D00)) AK; KIND(AK) =      8  
  
KIND(5.5D00) =      8  
  
A8 = 5.123456789123456789_8; PRINT A8      5.12345678912346  
A8 = 5.123456789123456789D00; PRINT A8      5.12345678912346  
A4 = 5.123456789123456789D00; PRINT A4      5.123457  
  
KIND(I2) - I2 tem quantos bytes?      2  
KIND(I4) - I4 tem quantos bytes?      4  
KIND(I8) - I8 tem quantos bytes?      8  
  
KIND(A4) - A4 tem quantos bytes?      4  
KIND(A8) - A8 tem quantos bytes?      8  
KIND(A16) - A16 tem quantos bytes?     16  
  
PRECISION(A4) - A4 tem quantos algarismos na mantissa?      6  
PRECISION(A8) - A8 tem quantos algarismos na mantissa?     15  
PRECISION(A16) - A16 tem quantos algarismos na mantissa?    33  
  
RANGE(A4) - A4 máximo expoente decimal?      37  
RANGE(A8) - A8 máximo expoente decimal?     307  
RANGE(A16) - A16 máximo expoente decimal?    4931  
  
Quero mantissa 20 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)      16  
Quero mantissa 10 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)      8  
Quero mantissa 05 alg. e exp.dec.020 - SELECTED_REAL_KIND(20,100)      4  
  
Quero inteiro exp.dec.100 SELECTED_INT_KIND(100)      -1  
Quero inteiro exp.dec.10 SELECTED_INT_KIND(10)      8  
Quero inteiro exp.dec.09 SELECTED_INT_KIND(09)      4  
Quero inteiro exp.dec.04 SELECTED_INT_KIND(09)      2  
  
I2 = 32000; PRINT I2      32000  
I2 = 33000; PRINT I2 - ERROR !!!
```

PGI

```
[xxxxxxx.xxxxxxx@s dumont14 ~]$ module load pgi/compilers-18.10
[xxxxxxx.xxxxxxx@s dumont14 ~]$ pgfortran lista2.f90
PGF90-W-0031-Illegal data type length specifier for real (lista2.f90: 7)
PGF90-W-0031-Illegal data type length specifier for a16 (lista2.f90: 7)
   0 inform,   2 warnings,   0 severes,   0 fatal for kindo
[xxxxxxx.xxxxxxx@s dumont14 ~]$ ./a.out
```

```
DOUBLE PRECISION A8 = 5.5_8      5.5000000000000000
DOUBLE PRECISION A8 = 5.5D00    5.5000000000000000
REAL A4 = 5.5_8      5.500000
```

```
REAL (KIND = 8) AJ; KIND(AJ) =      8
```

```
REAL (KIND = KIND(0.0D00)) AK; KIND(AK) =      8
```

```
KIND(5.5D00) =      8
```

```
A8 = 5.123456789123456789_8; PRINT A8      5.123456789123456
A8 = 5.123456789123456789D00; PRINT A8    5.123456789123456
A4 = 5.123456789123456789D00; PRINT A4    5.123457
```

```
KIND(I2) - I2 tem quantos bytes?      2
KIND(I4) - I4 tem quantos bytes?      4
KIND(I8) - I8 tem quantos bytes?      8
```

```
KIND(A4) - A4 tem quantos bytes?      4
KIND(A8) - A8 tem quantos bytes?      8
KIND(A16) - A16 tem quantos bytes?    4
```

```
PRECISION(A4) - A4 tem quantos algarismos na mantissa?      6
PRECISION(A8) - A8 tem quantos algarismos na mantissa?    15
PRECISION(A16) - A16 tem quantos algarismos na mantissa?    6
```

```
RANGE(A4) - A4 máximo expoente decimal?      37
RANGE(A8) - A8 máximo expoente decimal?    307
RANGE(A16) - A16 máximo expoente decimal?    37
```

```
Quero mantissa 20 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)    -1
Quero mantissa 10 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)      8
Quero mantissa 05 alg. e exp.dec.020 - SELECTED_REAL_KIND(20,100)      4
```

```
Quero inteiro exp.dec.100 SELECTED_INT_KIND(100)    -1
Quero inteiro exp.dec.10 SELECTED_INT_KIND(10)      8
Quero inteiro exp.dec.09 SELECTED_INT_KIND(09)      4
Quero inteiro exp.dec.04 SELECTED_INT_KIND(09)      2
```

```
I2 = 32000; PRINT I2      32000
I2 = 33000; PRINT I2 - ERROR !!!
```

FORTTRAN STOP

COMPARAÇÃO:

- Entre o gfortran e o iforth, a saída é praticamente a mesma, só variando a precisão do “PRINT A8” e “PRINT A4” ao rodar o programa.
 - Já com o pgfortran tivemos significativamente mais mudanças:
 - ao compilar tivemos “warnings” falando sobre “Illegal data type length specifier”;
 - A linha “KIND(A16) - A16 tem quantos bytes?” acusa 4 bytes (ao invés dos 16 dos outros compiladores);
 - A linha “PRECISION(A16) - A16 tem quantos algarismos na mantissa?” acusa 6 algarismos (ao invés dos 33 dos outros compiladores);
 - A linha “ RANGE(A16) - A16 máximo expoente decimal?” acusa 37 (ao invés dos 4931 dos demais);
 - E finalmente a linha “Quero mantissa 20 alg. e exp.dec.100 - SELECTED_REAL_KIND(20,100)” acusa -1 (ao invés de 16).
-

3)

ANÁLISE DAS SAÍDAS:

O gfortran e o ifort apresentaram saídas semelhantes, mostrando que devemos ter atenção mais com detalhes de precisão dos números de ponto flutuante.

O pgfortran mostrou que tem diferença significativa para o KIND A16.