シミュレーション物理3
プログラミングの基本　その2

大槻東巳

ohtsuki@sophia.ac.jpに出席確認のメールを出してください。件名に必ず，学生番号，氏名を書いてください。

Unix のコマンド

- whoami
- date
- who
- cal
- cal 07 1986 (1986年7月のカレンダー)
- man date

以上のコマンドをdahlmanにログインして、実際に試す

ディレクトリ（フォルダ）の移動

まず、unixの階層構造を理解しましょう

- pwdと打ってください
- /shome/……とでたのが皆さんの今いる場所です

cd public_html

pwd

で移動したことがわかります。

- 上の階層に行きたいときは

cd ..

ファイル操作

- コピー

cd
cp public_html/index.html index.html
移動したいときは
mv public_html/index.html index.html
フォルダ(ディレクトリ)を作る
mkdir test/
例　cp public_html/index.html test/index.html

ファイル操作2

cd test/

でtestというディレクトリに移動します

biwa

cp /home/work/test/test*.txt .

dahlman

cp /thome/rikou/gpscieng/test/text*.txt .

で共通directoryの中にあるtest*.txtを取ってくる。(最後の.を忘れないこと。.はここに持ってくるという意味。)

lsと打ち込んでみよう。ファイルが移動している。

ファイル操作3

- ls

でディレクトリ内のファイルの一覧が見える

ls –la

はファイルの詳しい属性を表示

rm filename

でファイルを消去できます。

注意)ファイルは消したら復活できません

Unix上での文書ファイルの操作

cat text1.txt

cat text2.txt

wc text1.txt (文字数や行数を計算)

more,head,tailを試す

diff text1.txt text2.txt （ファイルの比較）

grep -n critical text1.txt (critical を含んだ行)

ispell text2.txt (スペルチェック)

Emacsの使い方

- emacs filenameでファイルを開く，作る。
- 適当に編集して
- cntl-x cntl-sでセーブ
- cntl-x cntl-cで終了(ファイルが書き換えられていたら，セーブするか聞いてくる。)
- その他，cntl-sでサーチ。

プログラム中の変数

- 計算機はbitで処理。1byte=8bit
- 例：英数文字1byte=8bit=2**8(256通り)
  - 日本語などは2バイトで処理(2**16=65536通り)
- 整数は4 byte=32bit=2**32=4294967296
  - でも符号があるのでこの半分，-20億から+20億程度までしか使えない。（正確には-2**31から2**31-1まで。銀行などでは20億は小さすぎる。）
- 実数は?

実数の型

- 通常の実数は整数と同じく4バイト。しかし，指数部分も表現しなければいけないので，精度（有効数字）はせいぜい7桁。
- これでは高精度の計算は期待出来ない。
- 他に型はないか?

整数の型

integer	1	8bit, -2**7～2**7-1
integer	2	16bit, -2**15～2**15-1
integer	3	32bit, -2**31～2**31-1
integer	4	64bit, -2**63～2**63-1

実際の使い方

- 学生番号を整数で宣言，名前はstudent_id

integer(kind=2)::student_id

何桁の整数がkind=3とか覚えるのが面倒。
例えば10桁の整数を使いたいとき，このkindはいくつ
分からないものか?

Integer, parameter::digit10=selected_int_kind(10)
Integer(kind=digit10)::m,n

実数の型

- ３２ビット（７桁の有効数字）: 単精度
- ６４ビット（１４桁の有効数字）: 倍精度

Real(kind=1)::z　!単精度。(kind=1)は省略可

Real(kind=2)::z　!倍精度。(kind=2)は省略不可

例えば精度を１０桁以上でやりたい
Integer, parameter::precision10=selected_real_kind(10)
Real(kind=precision10)::a,b
有効数字の桁p,指数の大きさr(10**rということ)の場合，
Integer, parameter::precision10=selected_real_kind(10，４０)
Real(kind=precision10)::a,b

ここまで言ったら複素数

- Complex::a,b !これは単精度
- Integer,parameter::dp=selected_real_kind(14)

  complex(kind=dp)::gamma !倍精度

ついでに複素数の操作

Real(z):実数部分，Aimag(z):虚数部分，conjg(z):複素共役，cmplx(x,y,kind=2):x+iyを作る。

配列

- ３次元ベクトル：３成分。N次元:n成分
- 構文：例５次元配列

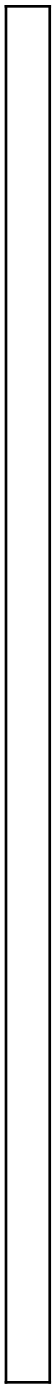
  Real(kind=dp),dimension(5)::a

  Real(kind=dp)::a(5)

  便利な配列関数：
  maxval(a),maxloc(a),minval(a),minloc(a),product(a),sum(a),dot_product(a,b)

```
program precision !プログラムのタイトル
!------------------------
!  This is a program to demonstrate the data types
!2005/4/25  Written by T. Ohtsuki
! ここにプログラムの変更履歴を書く。
! Modified by TO on 23rd Oct. 2009
!------------------------
implicit none  ! Always begin with this statement
integer,parameter::threedigit=selected_int_kind(3)
! parameter はプログラムの中で固定
integer,parameter::sixdigit=selected_int_kind(6)
integer,parameter::sp=selected_real_kind(5)
integer,parameter::dp=selected_real_kind(11,50)
integer(kind=threedigit)::i1,i2,i3
integer(kind=sixdigit)::j1,j2,j3
real(kind=sp)::x1,x2,x3
real(kind=dp)::y1,y2,y3

i1=200
i2=2000
i3=i1*i2
print *,i1,"times",i2, "with 16bit integers"
print *,i3  !print * は画面にかけという命令

j1=200
j2=2000
j3=j1*j2
print *,j1,"times",j2, "with 32bit integers"
print *,j3

x1=1000000.0001
x2=1000000.
print *, x1-x2

x1=4.0_sp
x2=atan(1.0_sp)
x3=x1*x2
print *,x1,"times",x2, "with 32bit real numbers"
print *,x3

y1=4.0_dp
y2=atan(1.0_dp)
y3=y1*y2
print *,y1,"times",y2, "with 64bit real numbers"
print *,y3


stop
end  !programがここで終わったことを示す。
```

```
program array
!------------------------
!  This is a program to demonstrate the array
usage
!2005/4/25  Written by T. Ohtsuki
!------------------------
implicit none  ! Always begin with this statement
integer,parameter::dp=selected_real_kind(11,50)
integer,parameter::n=3
integer::i,j
real(kind=dp),dimension(n)::y1,y2
real(kind=dp),dimension(n,n)::y3
complex(kind=dp),dimension(n)::y4



do i=1,n
y1(i)=1._dp/real(i)**2
y2(i)=real(i)**2
end do
print *, "sum of 1/i**2,i**2 are",sum(y1),sum(y2)
print *, "dot product of y1 and y2
is",dot_product(y1,y2)
```



```
do i=1,n
do j=1,n
y3(i,j)=i*j
end do
end do
print *,"2-d array (matrix) y3 is",y3

do i=1,n
y4(i)=cmplx(real(i),real(i)**2,kind=dp)
end do
print *,"Complex array y4 is ",y4
print *,"Cojugate of it is ",conjg(y4)

print *, "Multiplying the complex
numbers",y4(1),"times",y4(2),"is"
print *, y4(1)*y4(2)

print *,"exponential of ",y4(1),"is"
print *,exp(y4(1))



stop
end
```