

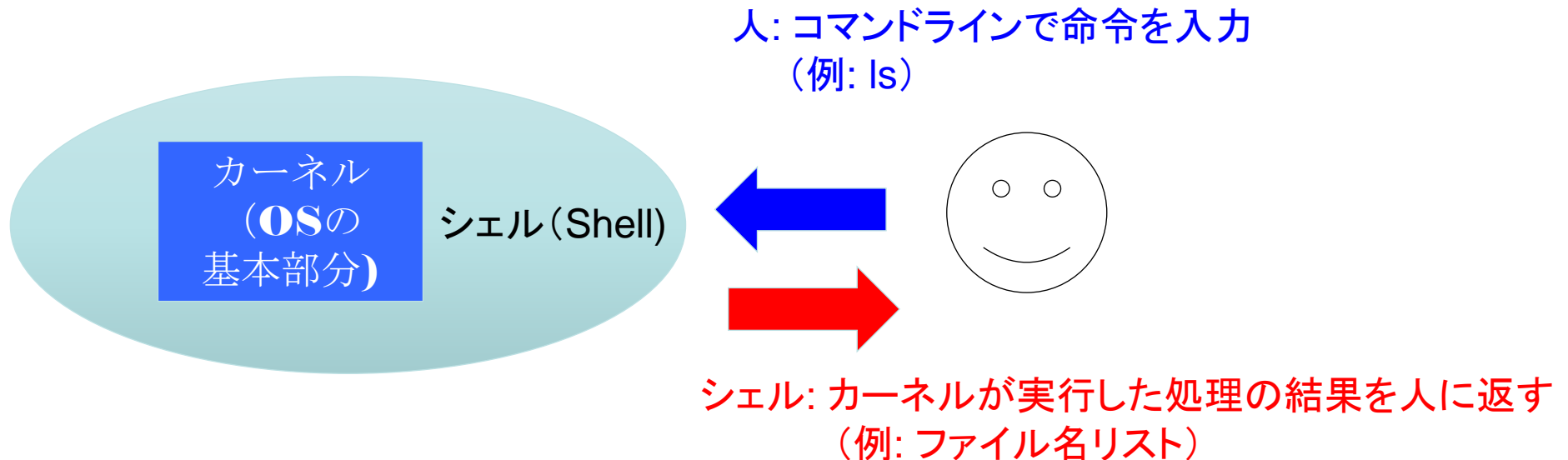
シェルスクリプト

森宙史 (**Hiroshi Mori**)

東京工業大学・生命理工

hmori@bio.titech.ac.jp

カーネルとシェルの関係



シェルは、コンピュータのカーネル(OS)と人とをつなぐ機能を持ったソフトウェア

シェルスクリプトとは

- コマンドを羅列したテキストファイルである。
 - 複数のコマンドを順番に実行できる。
- 「変数」、「コマンドライン引数」を使用することができる。

なぜシェルスクリプト

- 同じコマンドを何度も実行する必要がある時は、対話式の実行が面倒。
- **NGS**解析では、複数のコマンドを繋げて解析を行うことが多く、対話式の実行が面倒。
- 遺伝研スパコンを含む多くの計算環境では、シェルスクリプトで実行する必要がある。

シェルスクリプトの前に

- コマンドを実行してみましよう
 - **echo** 「与えられた文字列を表示する」
 - **sleep** 「指定された時間、停止する」
 - **date** 「日付や時刻を表示、設定する」

```
$echo start
```

```
$date
```

```
$sleep 5
```

```
$date
```

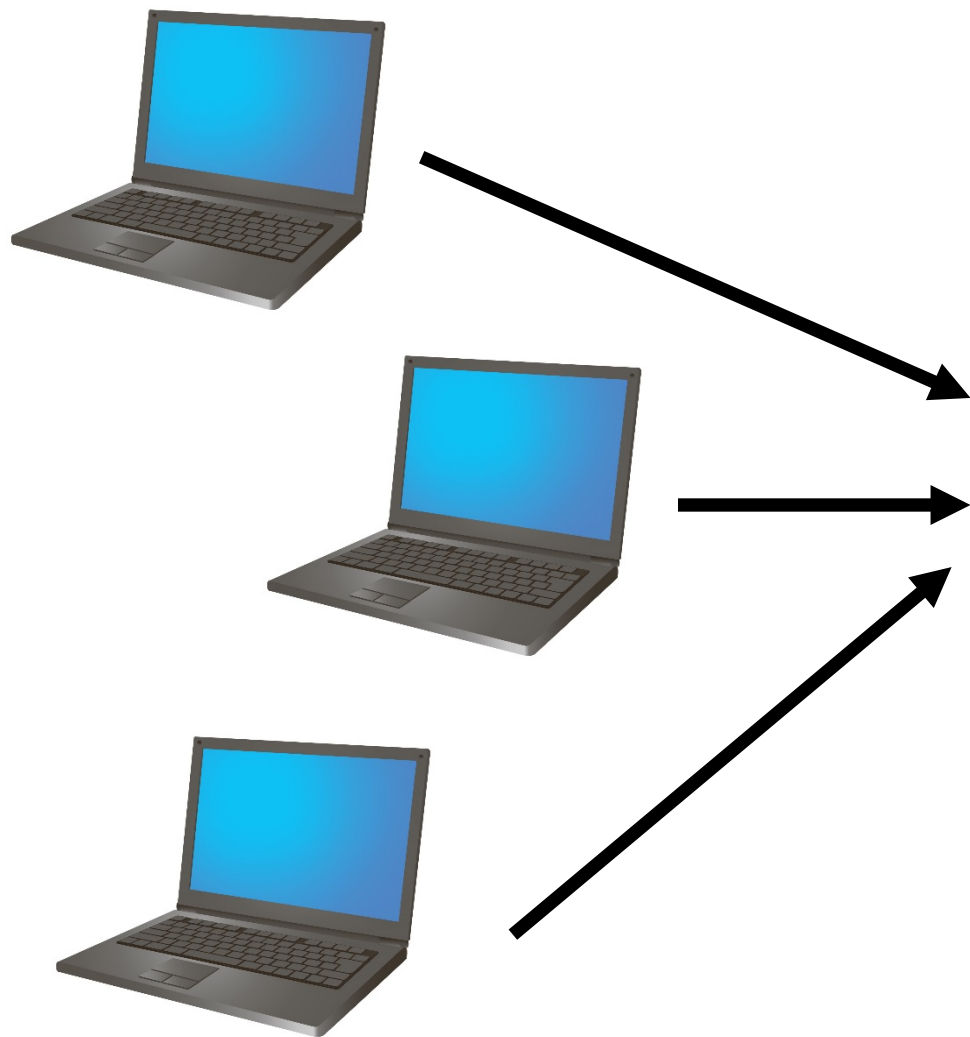
4つのコマンドを対話式で入力

コマンドライン上で困ったときは、”**Ctrl+c**”

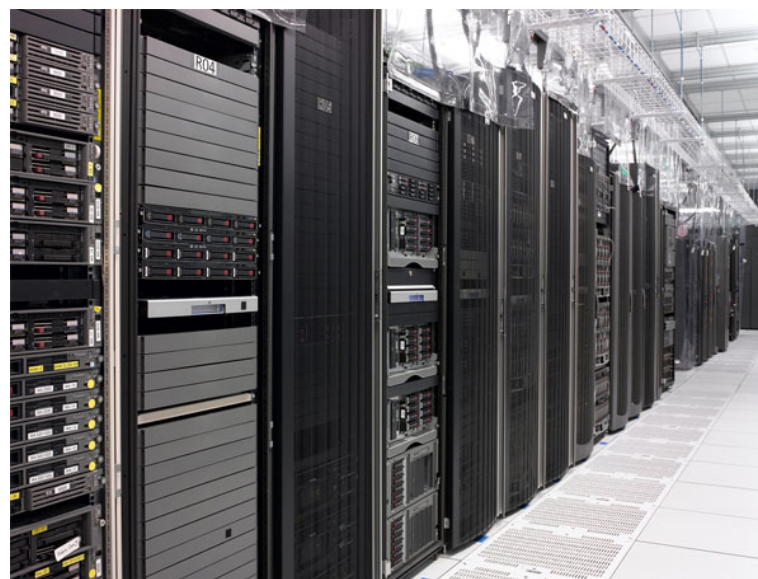
シェルスクリプトの実行

1. シェルスクリプトの作成

2. シェルスクリプトの実行

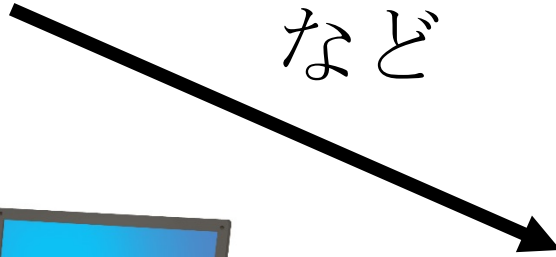


シェルフファイル

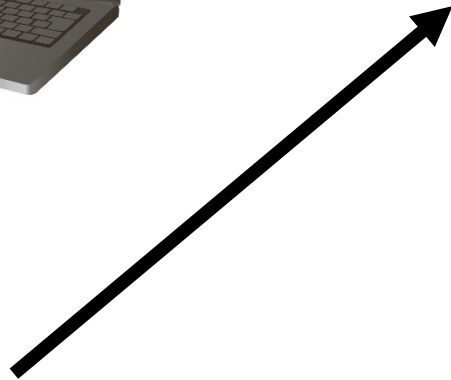


遺伝研スパコン

WinSCP ファイル転送 FileZilla など



シェルファイル



遺伝研スパコン

シェルスクリプトの実行

1. シェルスクリプトの作成

```
start.sh
```

```
1 |#!/bin/bash  
2 |echo start  
3 |date  
4 |sleep 5  
5 |date
```

bashを使うおまじない
echoコマンドを実行
dateコマンドを実行
sleepコマンドを実行
dateコマンドを実行

2. シェルスクリプトの実行

シェルスクリプトの実行

1. シェルスクリプトの作成

```
start.sh
```

```
1 |#!/bin/bash  
2 |echo start  
3 |date  
4 |sleep 5  
5 |date
```

bashを使うおまじない
echoコマンドを実行
dateコマンドを実行
sleepコマンドを実行
dateコマンドを実行

2. シェルスクリプトの実行

```
$chmod u+x start.sh  
$./start.sh
```

実行可能にする
シェルスクリプトを実行

- #以降は、コメントとなる
 - 実行時に無視されます。

```
start_rev.sh
```

```
1 | #!/bin/bash  
2 | # 2015 11 19  
3 | # author  
4 | echo start  
5 | date  
6 | sleep 5      #wait 5s  
7 | date
```

変数

- 変数は、文字・数値などを格納することができる。
 - 変数名=値 (“=”の前後のスペースは駄目)
 - **`${変数名}`**で呼び出す。

```
var1.sh
```

```
1 | #!/bin/bash  
2 | str="start"  
3 | echo ${str}
```

bashを使うおまじない
strに"start"を格納する
echoコマンドを実行

```
$chmod u+x var1.sh  
$./var1.sh
```

変数の展開

```
var2.sh
```

```
1 | #!/bin/bash  
2 | str="start"  
3 | echo str  
4 | echo `str`  
5 | echo "str"  
6 | echo ${str}  
7 | echo `${str}`  
8 | echo "${str}"
```

```
$chmod u+x var2.sh  
$./var2.sh
```

実行結果

```
str  
str  
str  
start  
`${str}`  
start
```

“”で囲むと変数が展開され、“`”で囲むと変数が展開されません。

変数 (コマンドライン引数)

```
var3.sh
```

```
1 | #!/bin/bash  
2 | str=$1  
3 | echo $1 str:${str}  
4 | echo $1 $2
```

\$1~\$n

シェルに与えられた引数の値。
\$1は第1引数、**\$2**は第2引
数...となる。

実行結果

```
$chmod u+x var3.sh  
$./var3.sh mouse  
$./var3.sh mouse human  
          ↑          ↑  
          $1          $2
```

mouse str:mouse
mouse

mouse str:mouse
mouse human

配列

- 値の集合を格納できます。
- スペース区切りで代入する値を指定します。

array1.sh

```
1 #!/bin/bash
2 ary=(1 2 X Y)
3 echo ${ary}
4 echo ${ary[@]}
5 echo ${ary[0]} ${ary[1]} ${ary[2]} ${ary[3]}
```

実行結果

1
1 2 X Y
1 2 X Y

- 処理を繰り返し実行したい場合は、**for文**（ループ処理）が便利です。

for文のかたち

```
for 変数 in 引数1 引数2 ...  
do  
    処理  
done
```

与えられた引数を変数に順番に代入し、引数がなくなり次第ループ処理を終了

for文

```
loop1.sh
```

```
1 #!/bin/bash  
2 for i in 1 2 X Y  
4 do  
5     echo chr_${i}  
6 done
```

実行結果

```
chr_1  
chr_2  
chr_X  
chr_Y
```

まとめ

- シェルスクリプトはコマンドの単なる羅列
- 新しいコマンドの作成が可能

【参考】

- ・ **UNIX & Linux** コマンド・シェルスクリプト
リファレンス

<http://shellscript.sunone.me/>

- ・ **Linux**標準教科書

<http://www.lpi.or.jp/linuxtext/text.shtml>

NIGスパコンでのqsubを用いたJobの実行

<https://sc.ddbj.nig.ac.jp/index.php/ja-howtouse>

or

遺伝研スパコン で検索

NIGスパコンでのqsubを用いたJobの実行

```
var1.sh
```

1	#!/bin/bash	bashを使うおまじない
2	#\$ -S /bin/bash	NIGスパコンでのおまじない
3	str="start"	strに"start"を格納する
4	echo \${str}	echoコマンドを実行

```
$qlogin
```

```
$qsub var1.sh
```

```
$qstat
```

```
$less var1.sh.o*
```