

情報処理概論

第2回 プログラムの翻訳と実行

情報基盤研究開発センター 谷本 輝夫

出席・課題提出状況の確認

- ▶ Web学習システムで確認
 - ▶ 左側メニューのツール → 成績表 で確認できる
 - ▶ 出席確認用のテストの得点を確認
 - ▶ 課題提出の状況を確認

今回の内容

- ▶ 前回の復習
- ▶ ファイルの操作
- ▶ プログラムの翻訳と実行
- ▶ 演習

前回の復習

- ▶ ログイン
- ▶ sshでahに接続する。
- ▶ 前回作成した test.f90 を再度 emacs で開く
- ▶ 中身を確認したら終了
- ▶ 新しく hello.f90 という名前でプログラムの作成開始
- ▶ 以下のプログラムを入力

```
program test
  write(*, *) 'My name is John.'
  write(*, *) 'Nice to meet you!'
  stop
end program
```

- ▶ 入力し終わったら、保存して終了

今回の内容

- ▶ 前回の復習
- ▶ ファイルの操作
- ▶ プログラムの翻訳と実行
- ▶ 演習

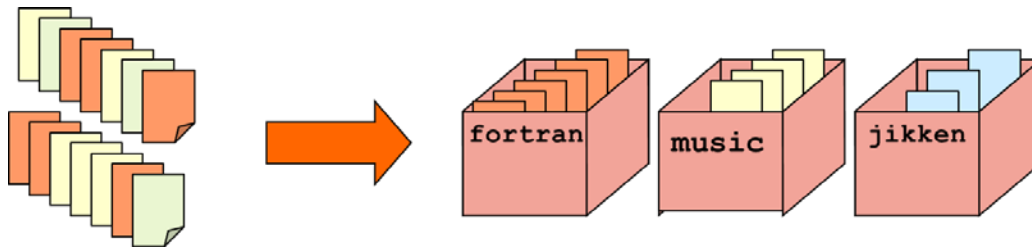
ファイルの操作

- ▶ ディレクトリ (=フォルダ) を使ってファイルを整理
- ▶ ファイル一覧を表示
- ▶ ファイルを移動する、ファイルの名前を変更
- ▶ ファイルをコピー
- ▶ ファイルを削除
- ▶ ファイルの内容を閲覧

ディレクトリ

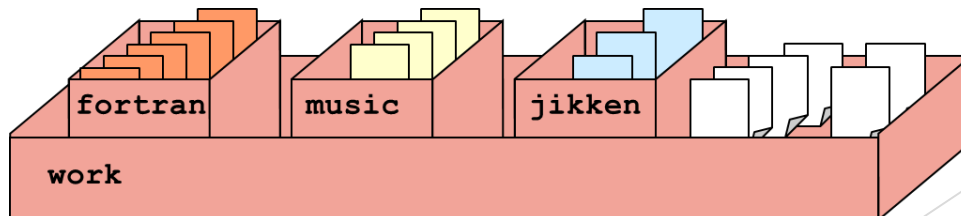
▶ ディレクトリ

- ▶ Windows や MacOS のフォルダとほぼ同じ
- ▶ ファイルの分類に利用



▶ ディレクトリの中にさらにディレクトリを作成可能

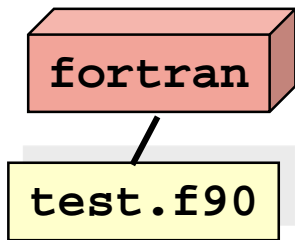
- ▶ フォルダと同様



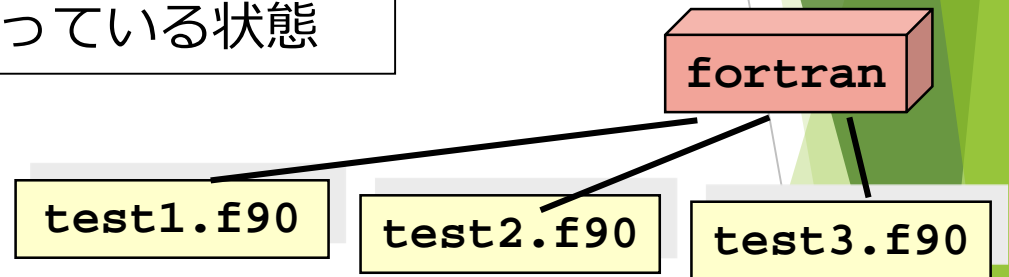
ディレクトリとファイル

- ▶ この講義ではディレクトリとファイルを以下の通り表現

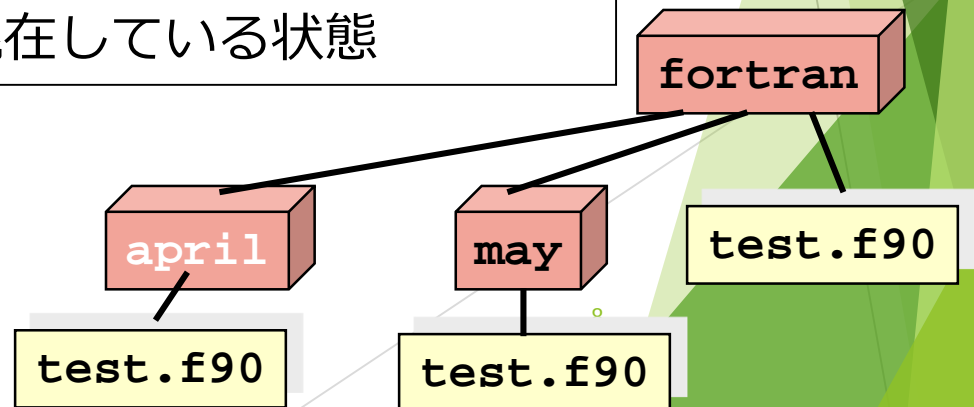
1つのディレクトリに
1つのファイルが
入っている状態



複数のファイルが
入っている状態

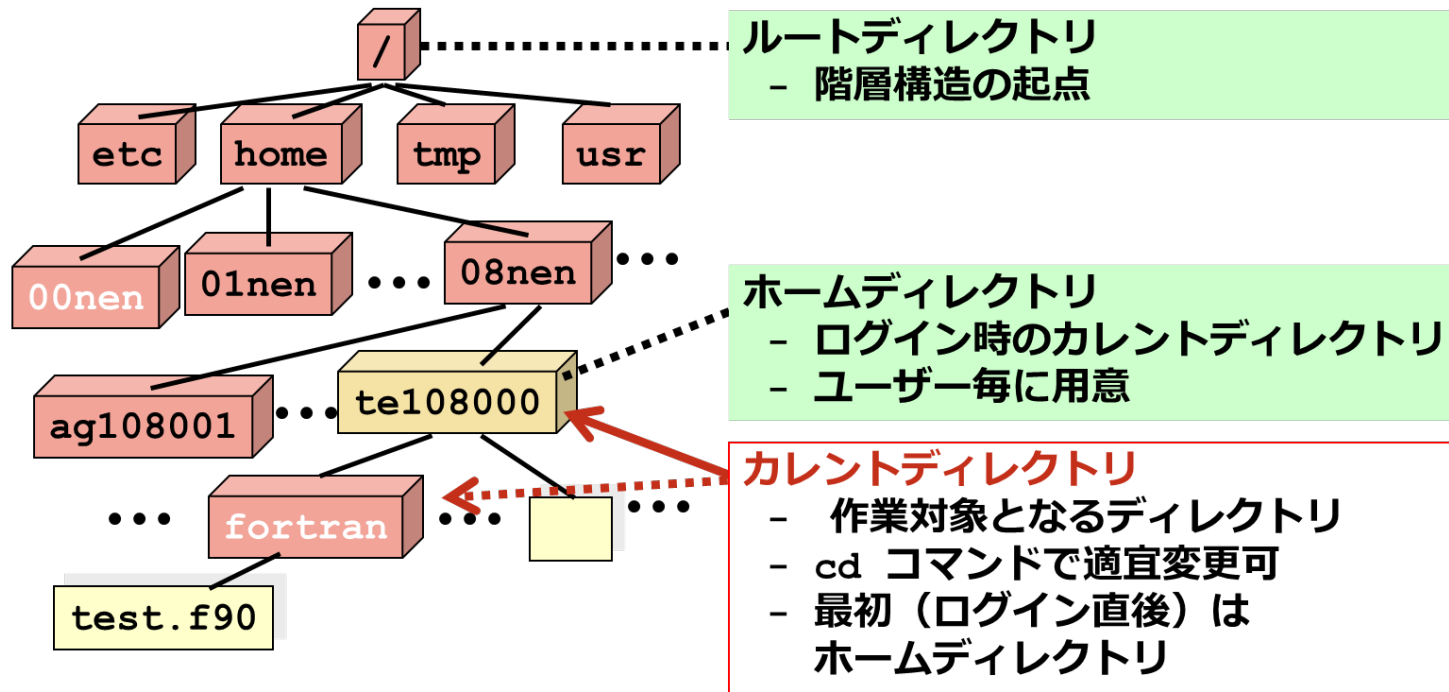


ディレクトリとファイルが
混在している状態



サーバ全体の構造

- ▶ 木を逆さまにしたような構造



カレントディレクトリとホームディレクトリ

- ▶ カレントディレクトリ（=現在の作業対象ディレクトリ）
 - ▶ そこを基準として、ファイルやディレクトリの場所を指定
- ▶ ログイン直後はホームディレクトリ（=自分専用）がカレントディレクトリ.
- ▶ 例1：カレントディレクトリの test.f90 というファイルを編集

```
$ emacs test.f90
```

- ▶ 例2：カレントディレクトリの test.f90 をカレントディレクトリの下に fortran ディレクトリに移動

```
$ mv test.f90 fortran
```

ディレクトリの作成

mkdir

- ▶ WindowsやMacOSでは、「フォルダの新規作成」

- ▶ 使い方 :

\$ mkdir <作成したいディレクトリの名前>

空白(スペース)を
忘れない

- ▶ 例 1 : fortranという名前のディレクトリを作成

```
$ mkdir fortran
```

- ▶ 例 2 : jikkenと picture という名前のディレクトリを作成

```
$ mkdir jikken picture
```

ファイルの移動

mv

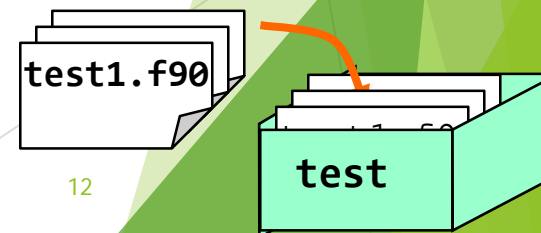
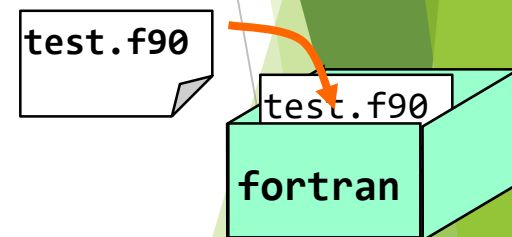
- ▶ 使い方 : mv (オプション) 移動元 移動先
 - ▶ 移動先がディレクトリならディレクトリへの移動
ファイルなら, ファイル名の変更

- ▶ 例 1 : test.f90 を fortranディレクトリに移動

```
$ mv test.f90 fortran
```

- ▶ 例 2 : test1.f90, test2.f90, test3.f90
を testディレクトリに移動

```
$ mv test1.f90 test2.f90 test3.f90 test
```



"カレントディレクトリ" の変更

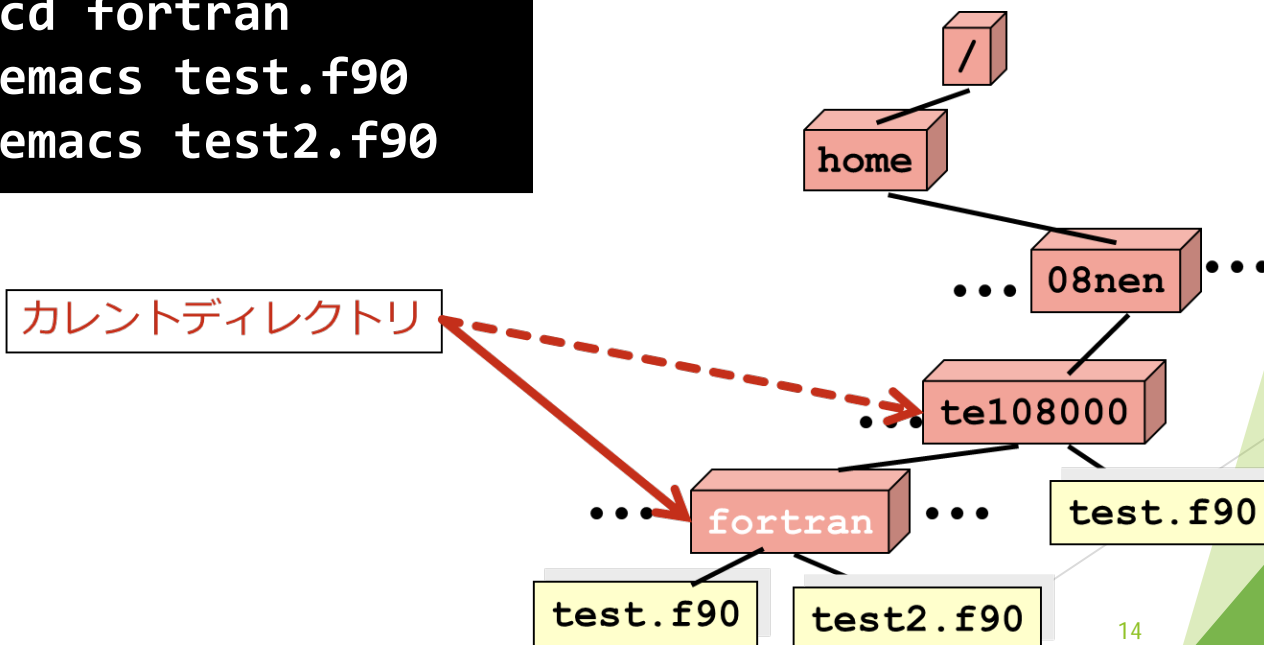
cd

- ▶ 使い方：
\$ cd <ディレクトリ名>
- ▶ ディレクトリ名を省略すると ホームディレクトリ に移動 **\$ cd**
- ▶ 例：fortranディレクトリに移動する
\$ cd fortran
- ▶ 特殊な表記：カレントディレクトリは“.”で表現
 - ▶ 一つ上のディレクトリは“..”で表現 **\$ cd ..**
- ▶ Windows や MacOS では、「フォルダをダブルクリック」

cd コマンドの利用例

- ▶ fortranディレクトリに移ってから、そのディレクトリにある test.f90 というファイルを編集し、さらに同じディレクトリにある test2.f90 というファイルを編集

```
$ cd fortran  
$ emacs test.f90  
$ emacs test2.f90
```



カレントディレクトリの表示

pwd

▶ 利用例 :

```
$ pwd
```

```
/home/08nen/te108000/fortran
```

pwdコマンドの実行結果 :

home ディレクトリの中の 08nen ディレクトリの中の
te108000ディレクトリの中の fortran ディレクトリ

▶ いつ使う ?

- ▶ 今どのディレクトリで作業しているかわからなくなったとき
- ▶ ファイルを削除する等の危険なコマンドを実行する前に確認しておきたいとき

ファイル一覧の表示

ls

- ▶ 指定したディレクトリの下の子ファイルやディレクトリ一覧を表示
- ▶ 使い方：
\$ ls (オプション) (<ディレクトリ名>や<ファイル名>)
- ▶ Windows や MacOS と違い、UNIX ではファイルの一覧が自動的に表示されない
- ▶ 利用例 1 : カレントディレクトリのファイル一覧

```
$ ls
```

- ▶ 利用例 2 : カレントディレクトリの下の子 fortran ディレクトリのファイル一覧

```
$ ls fortran
```


ls コマンドの便利な機能

- ▶ ファイルの詳細情報を見る： -l オプション追加
- ▶ 利用例： カレントディレクトリの test.f90 の詳細情報表示

```
$ ls -l test.f90  
-rw-r--r-- 1 2067047393 teacher 193 4月 22 22:35 test.f90
```

ファイルの所有者

ファイルのサイズ

最後に編集した日時

- ▶ ディレクトリの下の方まで辿って表示する
 - ▶ -R オプション追加
- ▶ 利用例：カレントディレクトリの下の方の全ファイルを表示

```
$ ls -R
```

ls コマンドで表示されるもの

- ▶ lsコマンドでファイルを見ると、色々なファイルが見れる

```
$ ls
test.f90 test.f90~ #test.f90#
```

普通ファイル emacsが残す前回のファイル emacsを正しく終了しなかった時に作成されるバックアップ

- ▶ 利用例：カレントディレクトリの test.f90 の詳細情報表示

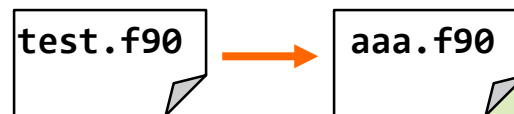
```
$ ls
Desktop Documents Favorites ...
$ ls -l
drwx----- 3 2167047393 teacher 4096 7月 14 10:28 2016 Desktop
drwx----- 18 2167047393 teacher 4096 7月 6 10:37 2015 Documents
drwx---r-x 4 2167047393 teacher 4096 7月 6 10:37 2015 Favorites
```

ファイルの名前変更

mv

- ▶ 使い方 : mv (オプション) 変更前 変更後
 - ▶ 移動先がディレクトリならディレクトリへの移動
ファイルなら, ファイル名の変更
- ▶ 利用例 : test.f90 の名前を aaa.f90 に変更

```
$ mv test.f90 aaa.f90
```



ファイルのコピー

cp

- ▶ 使い方 : cp (オプション) コピー元 コピー先
 - ▶ コピー元がファイルの場合、ファイルをコピー
 - ▶ コピー先がファイルなら、その名前のファイルとしてコピー
 - ▶ コピー先がディレクトリなら、コピー元と同じ名前のファイルとして、そのディレクトリ中にコピー
 - ▶ コピー元がディレクトリの場合、オプション -R を指定してディレクトリ全体をコピー
- ▶ 利用例 : test.f90 を test2.f90 という名前でコピー

```
$ cp test.f90 test2.f90
```

ファイルの削除

rm

- ▶ 使い方：
rm (オプション) 削除するファイルまたはディレクトリ ...
 - ▶ ディレクトリを削除する場合は, オプション-rを指定する.

- ▶ 利用例 1 : test.f90 と test2.f90 の削除

```
$ rm test.f90 test2.f90
```

- ▶ 利用例 2 : ディレクトリfortranの削除

```
$ rm -r fortran
```

ファイルの内容閲覧

less

- ▶ ファイルを編集せずに閲覧したい場合に利用
- ▶ 使用法 : less ファイル名
 - ▶ ファイルの内容を表示
 - ▶ 閲覧中は以下のキーを利用
 - ▶ 次のページへ : SPACEもしくは f
 - ▶ 前のページへ : b
 - ▶ 1行下へ : Enter もしくは j
 - ▶ 1行上へ : k
 - ▶ 閲覧終了 : q

ファイルの内容閲覧

cat

- ▶ 小さなファイルを編集せずに閲覧したい場合に利用
- ▶ 使用法 : cat ファイル名
 - ▶ ファイルの内容を表示して終了
 - ▶ (本来は入力を出力に繋ぐ為のコマンドで、指定がなければ標準入力(キーボード入力)を標準出力(モニター出力)に送る)
 - ▶ \$ cat <Enterキー> とすると、入力がそのままエコー(出力)される
 - ▶ 入力の終了は、C-d (Ctrlキーを押しながらd)
 - ▶ ファイル名を指定すると、入力がキーボードからファイルに切り替わる

演習

- ▶ fortran ディレクトリを作成
- ▶ カレントディレクトリの test.f90 と hello.f90 を fortran ディレクトリに移動
- ▶ カレントディレクトリを fortran ディレクトリに変更
- ▶ カレントディレクトリのファイル一覧を表示
- ▶ 現在のカレントディレクトリを表示

- ▶ 時間が余ったら、他にもいろいろ試してみる

演習の答え

- ▶ 以下の通り、コマンドを実行

```
$ mkdir fortran  
$ mv test.f90 hello.f90 fortran  
$ cd fortran  
$ ls  
$ pwd
```

主なUNIXのコマンドのまとめ

コマンド	動作
pwd	現在位置（カレントディレクトリ）の表示
ls	ファイル一覧の表示
less ファイル名	ファイルの内容表示
cd 移動先	カレントディレクトリの変更（移動）
mkdir ディレクトリ名	新規ディレクトリの作成
mv 移動元 移動先	ファイルの移動, ファイル名の変更
cp コピー元 コピー先	ファイルのコピー
rm ファイル名	ファイルの削除
emacs ファイル名	ファイルの新規作成, もしくは編集
gfortran ソースファイル -o 実行ファイル	Fortranプログラムの翻訳（コンパイル）
./実行ファイル名	プログラムの実行

ファイルの扱いに注意

- ▶ UNIXでは、ファイルを削除したら、本当に消える
 - ▶ 「ゴミ箱」が無い
- ▶ 以下の場合も、上書きされる側の内容が消える
 - ▶ コピー先に同じ名前のファイルがある
 - ▶ 移動先に同じ名前のファイルがある
- ▶ 用心のためのオプション (mv, cp, rmに共通) : -i
 - ▶ ファイル操作 (削除など) の際、確認してくれる。
 - ▶ エイリアスを設定しておく也确实
 - ▶ ホームディレクトリの .bashrc に以下を追加する。
 - ▶ alias rm='rm -i'
 - ▶ mv, cpについても同様に設定

今回の内容

- ▶ 前回の復習
- ▶ ファイルの操作
- ▶ プログラムの翻訳と実行
- ▶ 演習

人間向きの言語と 機械向きの言語

- ▶ プログラムには2種類ある
 - ▶ 高級言語（人間向きの言語）のプログラム
 - ▶ 機械語（機械向きの言語）のプログラム
- ▶ 高級言語：人間の言葉に近い言語
 - ▶ プログラムを読み書きしやすい表記
 - ▶ 例：Fortran, C, C++, Java, Ruby, ...
 - ▶ コンピュータは直接理解できない

```
program test
  write(*, *) 'Hello, Fortran'
stop
end program
```

- ▶ 機械語：機械が理解できる命令（の番号）による言語
 - ▶ 人間が直接読み書きすることは難しい
 - ▶ コンピュータが直接実行できる表記

```
457f 464c 0101 0001 0000 0000 0000 0000
0002 0003 0001 0000 a040 0804 0034 0000
057c 0005 0000 0000 0034 0020 0008 0028
0029 0026 0006 0000 0034 0000 8034 0804
8034 0804 0100 0000 0100 0000 0005 0000
0004 0000 0003 0000 0134 0000 8134 0804
8134 0804 0013 0000 0013 0000 0004 0000
0001 0000 0001 0000 0000 0000 8000 0804
8000 0804 91f8 0004 91f8 0004 0005 0000
1000 0000 0001 0000 91f8 0004 21f8 0809
...
```

プログラムの翻訳 = コンパイル

- ▶ 高級言語から機械語へ翻訳

ソースファイル

```
program test
  write(*, *) 'Hello, Fortran'
stop
end program
```



実行ファイル

```
457f 464c 0101 0001 0000 0000 0000 0000
0002 0003 0001 0000 a040 0804 0034 0000
057c 0005 0000 0000 0034 0020 0008 0028
0029 0026 0006 0000 0034 0000 8034 0804
8034 0804 0100 0000 0100 0000 0005 0000
0004 0000 0003 0000 0134 0000 8134 0804
8134 0804 0013 0000 0013 0000 0004 0000
0001 0000 0001 0000 0000 0000 8000 0804
8000 0804 91f8 0004 91f8 0004 0005 0000
1000 0000 0001 0000 91f8 0004 21f8 0809
...
```

- ▶ ソースファイル：高級言語のプログラムが記録されたファイル
 - ▶ emacs 等で作成、実行できない
- ▶ 実行ファイル：機械語のプログラムが記録されたファイル
 - ▶ コンパイルによって作成、実行できる

Fortranプログラムのコンパイル

gfortran

- ▶ 使い方 : gfortran ソースファイル -o 実行ファイル
指定しない場合 a.out
- ▶ 例 : test.f90 をコンパイルして, その翻訳結果の機械語プログラムを test という名前のファイルに保存する

```
$ gfortran test.f90 -o test
```

ソースファイルの名前は .f90 にする

- ▶ 本講義で扱う Fortran90 のプログラムを作成する際、ファイル名の末尾を必ず .f90 とする

```
$ emacs test.f90
```



例) test.f90 , sample.f90 , a.f90等

- ▶ 拡張子：ファイル名の末尾に付ける文字列
 - ▶ ファイルの種類への識別に（人間が / システムが）利用
 - ▶ 拡張子が違えば正しく動作しない場合もある。
 - ▶ 他の拡張子の例：
 - ▶ .html, .htm HTML形式ファイル（Webページ）
 - ▶ .c C言語等

プログラムの実行

- ▶ コンパイルによって得られた実行ファイルの名前をコマンドとして利用する
- ▶ 例：カレントディレクトリの実行ファイル test の実行

```
$ ./test
```

- ▶ UNIXのコマンドと区別するために実行ファイルの前に必ず ./ を付ける

今回の内容

- ▶ 前回の復習
- ▶ ファイルの操作
- ▶ プログラムの翻訳と実行
- ▶ 演習

演習

- ▶ カレントディレクトリをホームディレクトリに変更
- ▶ カレントディレクトリを fortranディレクトリに変更
- ▶ ファイルの一覧表示
- ▶ hello.f90 をコンパイルし, 翻訳結果を helloという名前のファイルに格納
- ▶ 作成された hello を実行

演習の提出方法

1. 作成したファイル hello.f90 を Moodleの [ファイル提出] から提出
2. 実行の様子 of 提出 (報告)
 - ▶ ターミナル上の文字列 (コマンド実行の様子) をマウスのドラッグにより選択する
 - ▶ Moodleの提出画面にペーストする
windows: [Ctrl]を押しながらv or 右クリック→貼り付け(P)
mac: [command]を押しながらv or 2本指タップ→ペースト
3. アンケート

Emacs の練習

1. Emacs を開く
2. Emacs tutorial を実行 C-h t
3. 後は Emacs のチュートリアルに従う