実際にやってみよう

以降より，昨年度の講義で使用されたスライドをそのまま転載します．以降は参考までとしてください．

学科サーバへのログイン

- パスワードを入手：

  https://islay.ci.seikei.ac.jp/myshell/prog.cgi
- ログインのためにWebブラウザでアクセス:

  https://cisvr.ci.seikei.ac.jp/prog/login/
  - Firefox, Chrome, EdgeはOK, safariはNG
- `ciprog login:` でus202xxxのアカウント名を入力
- `us202xxx@ciprog's password:` でパスワードを入力
  - 入力パスワードは表示されないので注意

UnixのCUI

- CUI: Character(or Command) User Interface
- CLI: Command-Line Interface
- シェルプログラムとの対話
  - OSの一番外側の殻（shell)をなすプログラム

1. プロンプト（入力促進）の表示
2. コマンド文字列の入力
3. 文字列に対応するプログラムを探して実行
4. 実行結果の出力（何も表示しない場合もある）
5. 1へもどる

```
$               <<---- プロンプト
$ ls            <<---- 入力して実行
abc.cpp
$
```

プログラムファイルの作成（emacs）

- `emacs` コマンドとソースファイル名を指定して開始

```
$ emacs hello.cpp
```

- ファイルが開いたら，プログラムを入力
  - 矢印キーでカーソルの移動, 英数字記号キーで入力
  - 右表のコマンドを使用し，ファイルの保存などする

コマンド	できること
`C-x C-c`	emacsの終了
`C-x C-s`	ファイルの上書き保存
`C-z`	emacsの一時中断（復帰はfgコマンド）
`C-g`	何かを間違えた時

- `C-x`：Ctrlキーを押したままxのキーを同時にタイプ
- `C-x C-s`：`C-x` をタイプ．その後一度キーを離して，`C-s` をタイプ．
  - Ctrlキーは押したままでも良い

コンパイルと実行:方法1

- 毎回emacsを終了する場合（時間がかかる）

```
$ emacs hello.cpp
<プログラムを入力>
<C-x C-s でファイルを保存>
<C-x C-c でemacsを終了>
$ g++ -std=c++17 hello.cpp
$ ./a.out                    コンパイル成功ならば
hello, world

$ emacs hello.cpp              再度emacsを起動
```

コンパイルと実行:方法2

- emacsを一時停止する場合（素早い）

```
$ emacs hello.cpp
<プログラムを入力>
<C-x C-s でファイルを保存>
<C-z でemacsを一時停止>
[1]+  停止                  emacs hello.cpp
$ g++ -std=c++17 hello.cpp
$ ./a.out                  コンパイル成功ならば
hello, world

$ fg                       emacsの表示にもどる
```

注意：`fg`をし忘れると停止したemacsが残る
`jobs`コマンドで残ったemacsを確認できる

Unixの基本コマンド

コマンド	意味
ls	ディレクトリの内容をリスト表示
cat ファイル名	ファイルの中身を画面に出力
pwd	現在の作業ディレクトリ名の表示
cd ディレクトリ名	作業ディレクトリの変更
mkdir ディレクトリ名	ディレクトリの作成
cp ファイル名 名前	ファイルのコピー
mv 名前1 名前2	名前の変更またはファイルの移動
rm ファイル名	ファイルの削除
rmdir ディレクトリ名	ディレクトリの削除

- 特別な名前によるディレクトリ指定
  - 現在の作業ディレクトリ　：`.`（ドット）
  - 親ディレクトリ　：`..`（ドットドット）

実行例

```
$ pwd                          現在の作業ディレクトリ名を表示
/home/us202xxx
$ ls                           ファイルの一覧
a.out hello hello.cpp
$ rm a.out hello               ファイルの削除
$ ls
hello.cpp
$ mkdir c++prog1               ディレクトリを作成
$ ls                           作成できたかを確認
c++prog1 hello.cpp
$ mv hello.cpp c++prog1        ファイルの移動
$ ls                           移動を確認
c++prog1
$ ls c++prog1                  ディレクトリ名を指定して表示
hello.cpp
$ cd c++prog1                  作業ディレクトリを変更
$ pwd                          現在の作業ディレクトリ名を表示
/home/us202xxx/c++prog1
$ cp hello.cpp hello2.cpp      ファイルのコピー
$ ls                           コピーできたかを確認
hello.cpp hello2.cpp
$ cd ..                        親ディレクトリに移動
$ pwd                          移動を確認
/home/us202xxx
```

実行結果の提出

- `script` コマンド: コマンド実行の履歴をファイルに保存

```
$ cd c++prog1
$ pwd
/home/us202xxx/c++prog1
$ script
スクリプトを開始しました。ファイルは typescript です。
$ ls
hello.cpp hello2.cpp typescript
$ cat hello.cpp
....ファイルの内容を表示
$ g++ -std=c++17 hello.cpp
$ ./a.out
hello, world

$ exit
スクリプトを終了しました。ファイルは typescript です。
```

- `Jupyter Notebook` で `typescript` をダウンロード
- CoursePower にアップロード