

# C プログラミング 入門

— Unix コマンド —

早稲田大学

## 今回の目標

- 基本的な Linux コマンドを習得する  
man, pwd, ls, cd, mkdir, rmdir, cp, mv, rm, less, chmod
- エディタの基本操作ができるようになる  
Emacs

# 基本的な Unix コマンド

man	各コマンドのヘルプを表示
pwd	現在いるディレクトリを表示
cd	ディレクトリを移動
ls	現在いるディレクトリの中のファイル一覧を表示
mkdir	ディレクトリの作成
rmdir	ディレクトリの削除
cp	ファイルのコピー
mv	ファイル名の変更, ファイルの移動
rm	ファイルの削除
less	ファイルの中身を表示
chmod	アクセス権を変更
<b>【 】【 】</b>	入力したコマンドの履歴の表示
<b>【Tab】</b>	ファイル名・ディレクトリ名の自動補完

## Unix コマンド -man-

man: (manual) 各コマンドのヘルプを表示

自分自身のヘルプを表示させる

```
$ man man
```

- 【**↑**】【**↓**】や【Pg Dn】【Pg Up】で操作
- 終了は【q】

# Unix コマンド -pwd-

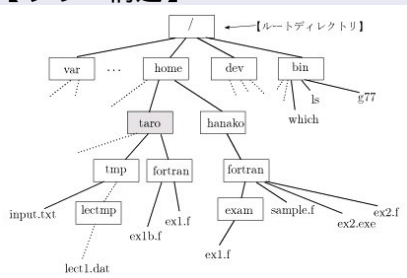
pwd:(print working directory) 現在いるディレクトリを表示

現在の場所を表示させる

[~]\$ pwd

- ディレクトリはツリー構造
- 各ディレクトリの中にファイルがある
- 端末起動時，最初にいる場所が自分のホームディレクトリ

## 【ツリー構造】



## Unix コマンド -ls-

ls : (list) 現在いるディレクトリの中のファイル一覧を表示

現在のディレクトリの中身を表示させる

```
[~] $ ls
```

現在いるディレクトリの中身を表示せよ : **ls**

# Unix コマンド -ls-

オプションをつけて、詳しい内容を表示させられる。

詳しい内容を表示

```
[~] $ ls -al
```

すべてのファイルを表示

```
[~] $ ls -a
```

ディレクトリ「Desktop」の中身を表示させる

```
[~] $ ls Desktop
```

他のオプションをヘルプを用いて調べよ：[man ls](#)

## Unix コマンド -mkdir-

mkdir:(make directory) ディレクトリの作成

work ディレクトリを作成する

```
[~] $ mkdir work
```

work ディレクトリができていることを確認せよ : **ls**



## Unix コマンド -rmdir-

rmdir:(remove directory) ディレクトリの削除

work ディレクトリを削除する

```
[~] $ rmdir work
```

work ディレクトリがなくなっていることを確認せよ : **ls**

## Unix コマンド - 【 】 -

【 】: 1つ前のコマンドを自動入力(戻しすぎたら【 】)

もう一度 work ディレクトリを作成する

[~] \$ 【 】(mkdir work でも可)

work ディレクトリを作成せよ : **mkdir work**

## Unix コマンド - 【Tab】 -

### 【Tab】: ファイル名・ディレクトリ名の自動補完

もう一度 work ディレクトリを削除

```
[~] $ rmdir w【Tab】
```

- ディレクトリ名を途中まで入力して【Tab】を押せば、すべてを入力する必要がない。
- ファイル名・ディレクトリ名で大文字・小文字が区別される。

## Unix コマンド -cd-

cd : (change directory) ディレクトリを移動する

work ディレクトリに移動する

```
[~] $ cd work
```

```
[~/work] $
```

現在いるディレクトリを表示させよ: `pwd`

## Unix コマンド -cd-

親ディレクトリに移動する

```
[~/work] $ cd ..
```

```
[~] $
```

- 「..」は親ディレクトリを、「~」はホームディレクトリを表す。
- cd の後にはスペースが必要

# Emacs

Emacs を起動させる

```
[~/work] $ emacs &
```

- テキストファイルを作成・修正できる
- 今後は、Emacs を利用して C 言語のプログラムファイルを作成する（慣れている方は「vi」、「gedit」等でも可）
- 今後作るファイルはすべて work ディレクトリに作成する
- 「&」は端末を使えるままにしておくため（Firefox と同じ理由）
- 終了させるためには「**【Ctrl】 + 【x】** **【Ctrl】 + 【c】**」

# Emacs

ファイル名を指定して起動する

```
[~/work] $emacs hello.c &
```

- 指定したファイルが存在しなければ新規に作成され、存在すればそのファイルが開かれる。

# Emacs

ファイル名を指定して起動する

```
[~/work] $emacs hello.c &
```

- 適当に文字を打ち込んで保存 ( **【Ctrl】 + 【x】** , **【Ctrl】 + 【s】** ) 後 , Emacs を終了する .
- 「半角」(英数字) と 「全角」(日本語) の切り替えは **【Ctrl】 + 【¥】**
- 必要なキー操作は各自覚えること .

hello.c ファイルが作成されていることを確認: **ls**



## Unix コマンド -less-

### less ファイルの中身を表示

hello.c ファイルの中身を表示させる

```
[~/work] $ less hello.c
```

- 打ち込んだ中身が表示されていることを確認
- ファイル内の移動は【 】【 】や【Pg Down】【Pg Up】で操作
- 【q】で終了

## Unix コマンド -cp-

cp : (copy) ファイルのコピー

hello.c のコピーを sample.c という名前で作成する

```
[~/work] $ cp hello.c sample.c
```

sample.c が作成されていることを確認: `ls`

中身が同じであることを確認 : `less sample.c`

## Unix コマンド -cp-

違うディレクトリにコピーする場合

```
[~/work] $ cp hello.c ../temp.c
```

- hello.c ファイルを親ディレクトリに temp.c という名前でコピーする

親ディレクトリへ移動 : `cd ..`

temp.c が作成されていることを確認 : `less temp.c`

work ディレクトリに移動する : `cd work`

## Unix コマンド -mv-

mv : (move) ファイル名の変更 , ファイルの移動

sample.c ファイルを temp2.c という名前に変更する

```
[~/work] $ mv sample.c temp2.c
```

sample.c がなくなったことを確認 : **ls**

## Unix コマンド -mv-

temp2.c を親ディレクトリに temp3.c という名前で移動

```
[~/work] $ mv temp2.c ../temp3.c
```

temp2.c がなくなったことを確認 : `ls`

親ディレクトリに移動し , temp3.c があることを確認 : `cd .. , ls`

## Unix コマンド -rm-

### rm : (remove) ファイルの削除

temp.c を削除する

```
[~] $ rm temp.c
```

- 一度削除したファイルは元に戻せないので注意が必要

temp.c がなくなったことを確認 : `ls`

## Unix コマンド -rm-

temp3.c ファイルを確認付きで削除する

```
[~] $ rm -i temp3.c
```

```
rm: 'temp3.c' を削除しますか (yes/no)? y
```

- オプション `-i` をつけると削除の確認ができる
- `y` を入力して【Enter】で削除され、`n` で削除されない。

work ディレクトリに移動する : `cd work`

## Unix コマンド -chmod-

chmod : (change mode) アクセス権を変更

hello.c の詳細情報を見る

```
[~/work] $ ls -l
```

```
-rw-r--r--  1 1W120000student  89 Apr 23 19:02 hello.c
```

- 自分で作成したファイルは，他人でも読むだけなら可能
- hello.c の詳細情報を見る
- 「-rw-r--r--」の最初の rw は自分で読める (r), 書ける (w) ことを表す
- 2 番目の r はグループの人の権利を表している．
- 3 番目の r はそれ以外の人々の権利を表している．



## Unix コマンド -chmod-

他人からファイルを読めなくするためには

```
[~/work] $ chmod go-r hello.c
```

- グループの人 (g) とそれ以外の人 (o) から読む権利 (r) をなくしている .

「-rw——」になったことを確認せよ : `ls -l`

## Unix コマンド -chmod-

他人から読めるようにするには

```
[~/work] $ chmod go+r hello.c
```

- 今後、他人に自分のファイルを見られたくない場合は、適宜アクセス権を変更すること。
- メール関係のファイルは最初から他人が見られないようになっている。

「-rw-r-r-」になったことを確認せよ: `ls -l`

# まとめ

- 基本的な Linux コマンド  
man, pwd, ls, cd, mkdir, rmdir, cp, mv, rm, less, chmod
- エディタの基本操作 Emacs