

電算機演習 1 2006 第 1 週目

池口 徹

埼玉大学 大学院 理工学研究科 情報数理科学専攻

338-8570 さいたま市 桜区 下大久保 255

Tel : 048-858-3577, Fax : 048-858-3716

Email : tohru@ics.saitama-u.ac.jp

URL : <http://www.nls.ics.saitama-u.ac.jp/~tohru>

今日の講義は？

- 機器について確認する
- Unix の使い方を学ぶ
 1. ログイン , ログアウト .
 2. Unix の仕組みを知る .
 3. Unix のコマンドを知る , 使う .

使用機器の確認

- **コンピュータ機器**
 - ディスプレイ
 - システム装置
 - キーボード
 - マウス
 - プリンタ

- **付属機器**

キーボード

- 記号と読み方
- 以下のキーの位置を確認
 - Control (Ctrl)
 - Shift
 - Caps Lock
 - Enter
 - Delete
 - BackSpace

マウスの使用法

- 3つのボタン
 - 左
 - 右
 - ホイール
- マウスの裏
- クリック
- ダブルクリック
- ドラッグ

Windows へログオン

- ログオン
- ログオフ
- パスワードとセキュリティ
 - パスワードの扱い方
 - パスワードの変更
 - パスワードの規制
 - 悪いパスワード
 - 良いパスワード

参考資料について

- **コンピュータ利用の手引 2006 年度版**
 - 1. **PART II Windows 編**
 - **第 4 章 UNIX の利用法**
 - 2. **PART III UNIX 編**
 - **第 1 章 UNIX(AIX) の簡単な利用法**
 - **第 3 章 UNIX の仕組み**
 - **第 4 章 UNIX コマンド**
 - **第 5 章 Shell**
 - **第 6 章 Emacs の使用法**
 - **第 8 章 L^AT_EX2e の使用法**
 - **第 13 章 UNIX 環境設定について**
 - **第 14 章 トラブル時の対処法**

Unix の特徴

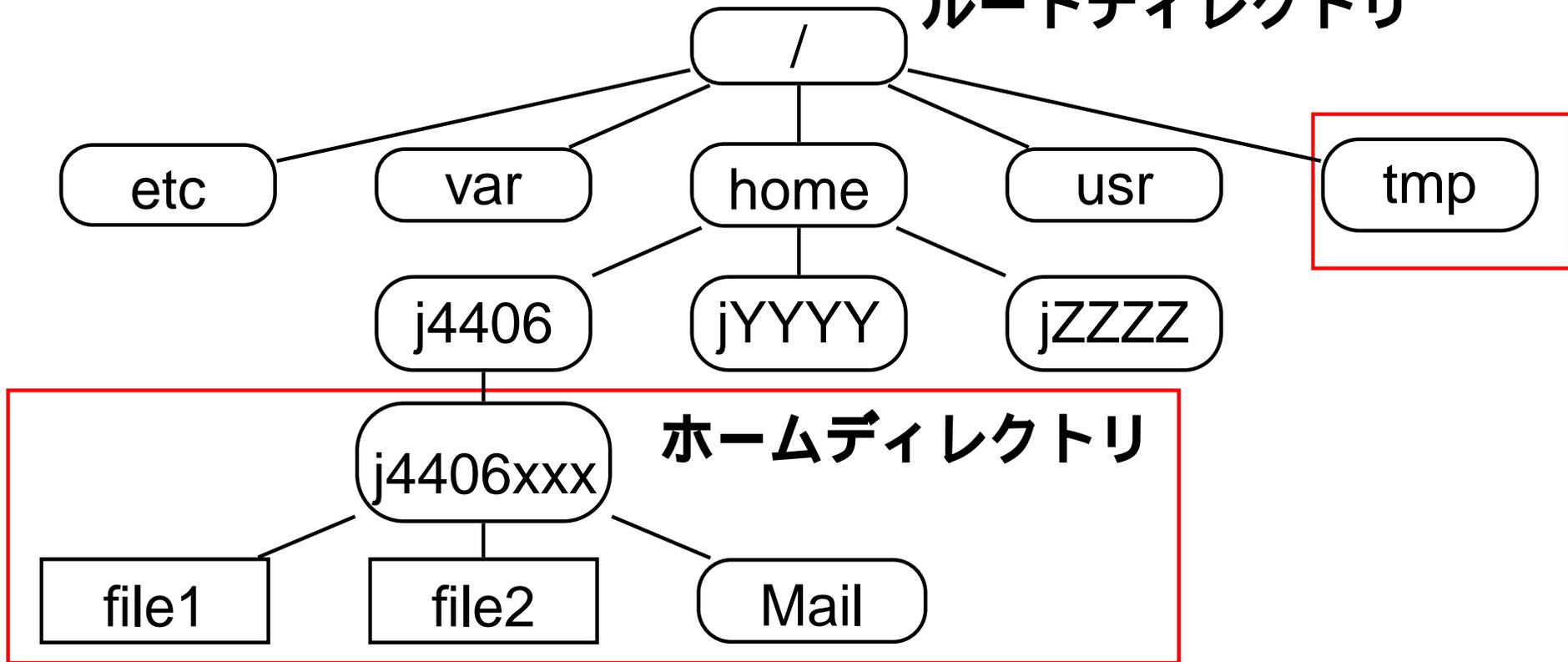
1. マルチ・ユーザー，マルチ・プロセス．
複数のユーザーが複数のプロセスを同時に走らせる
2. 単純なコマンド群を組み合わせて，複雑なプログラムを作成．
 - 標準入出力
 - パイプ
 - ファイル・ディスクリプター、ソケット
3. シェル (柔軟なユーザー・インターフェース)
 - コマンド・インタープリターとしてのシェル
 - プログラム言語としてのシェル
 - いろいろなシェル
4. 階層的なファイル・システム。
 - ファイルとディスクリプター
 - マウントによる複数のファイル・システムの統合
 - リモート・マウント
 - 周辺機器

ログインとログアウト

- とりあえず、ログインしてみましょう。
 - ログイン名
 - パスワード
- ログアウトも出来るかどうか、確認!

ディレクトリ

ルートディレクトリ



- ❑ 書き込みが出来るのは、自分のホームディレクトリと /tmp のみ。

ディレクトリ関連の読み方・呼ばれ方

- ❑ / (root directory) ルート
“/” 自体は slash スラッシュ → スラ
- ❑ /var (variable) ヴァー
- ❑ /etc (et cetera) エトセ
- ❑ /dev (device) デヴ
- ❑ /home (home directory) ホーム
- ❑ /tmp (temporary) テンプ
- ❑ /usr (user) ユーザ
- ❑ /usr/bin (binary) ユーザ・ビン
- ❑ /usr/lib (library) ユーザ・リブ
- ❑ /usr/local/bin (binary) ユーザ・ローカル・ビン
- ❑ /usr/share/man (manual) ユーザ・シェア・マン

これから何を学ぶか

- ❑ どのようなファイルが存在してるかを見る . (ls)

- ❑ ディレクトリファイルを作る・消す (mkdir, rmdir)
- ❑ ディレクトリ間を移動する (cd)

- ❑ ファイルをコピーする (cp)
- ❑ ファイルを消す (rm)
- ❑ ファイルを移動 , ファイルの名前を変える (mv)

- ❑ ファイルの中味を見る (cat, more, less)

- ❑ ファイルを新たに作る
- ❑ ファイルの内容を書き換える

- ❑ ファイルを処理する
 - 文章のファイル
 - ホームページのファイル
 - プログラムのファイル

ファイルを見る

注意 % はプロンプト

□ ls コマンドの使い方

% ls [オプション] [ファイル名]

% ls -a

% ls -l

% ls -alF

% ls -d

演習 1 ログインした直後の状態で、どのようなファイルが存在しているか確認しなさい。

演習 2 ls にオプション -a をつけた場合とつけない場合との違いは何か。

演習 3 ls にオプション -l をつけた場合とつけない場合との違いは何か。

ファイルの属性 (p.149)

```
% ls -la
```

とすると、ファイルの属性情報が表示される。

- ❑ 名前
- ❑ 所有者
- ❑ 大きさ
- ❑ 書き込み許可権

例

```
drwx--x--x 1 jXXXXXXXXX edu 256 Mar 24 19:12 .im
-r-xr-xr-x 1 jXXXXXXXXX edu 43 May 19 14:20 .cshrc
drwx----- 1 jXXXXXXXXX edu 512 Feb 24 11:48 Mail
```

許可権

□ 誰に対する許可があるか

- 本人 (所有者) user
- グループ group
- それ以外 others

□ どのような許可があるか

- 読み出し read
- 書き込み write
- 実行 execute

□ 例

`drwxr-x---`

ディレクトリ, 本人 `rw`x 可, グループ内 `rx` 可, 他不可

`-rwxr--r--`

ファイル, 本人 `rw`x 可, グループ内 `r` のみ可, 他 `r` のみ可

ディレクトリを作ってみよう

□ コマンド

- **作る** mkdir (make directory **の略**)
% mkdir **ディレクトリ名**
- **消す** rmdir (remove directory **の略**)
% rmdir **ディレクトリ名**

演習 4 各自のホームディレクトリ直下に 2006 という名前のディレクトリを作りなさい。また、そのディレクトリを消しなさい。

ディレクトリが作られたか、消されたかを確認するには、ls コマンドを用いるとよい。

ディレクトリ間を移動してみよう

- **どのディレクトリにいるかを表示**
% pwd (⇐ print working directory **の略**)
- **ディレクトリ間移動 cd (change directory **の略**)**
 - **特定のディレクトリに移る**
% cd **ディレクトリ名**
% cd /tmp
% cd /home/j4406
% cd /etc
 - **今いるディレクトリの上のディレクトリに移る**
% cd ..
% cd ../..
% cd ../../..
 - **ホームディレクトリに戻る**
% cd

ディレクトリ間を移動してみよう

演習 5 先ほど作成したディレクトリ 2006 に移動

演習 6 2006 において, 新たに四つのディレクトリ
apr, may, jun, jul, aug を作成

演習 7 ls, ls -la, pwd などで常に確認

演習 8 aug を削除

演習 9 apr に移動

演習 10 apr から may に, may から jun に移動

演習 11 自分のホームに移動

演習 12 apr に移動

演習 13 apr から一つ上, 二つ上, 三つ上のディレクトリに移動

ファイルを扱う

演習 14 ディレクトリ apr に移動

演習 15 以下の命令を実行したあと，どのようなファイルが出来ているかを確認しなさい．

```
% mksamp cmd
```

```
% ls -la
```

□ 以下のファイル名を確認できますか？

- text.EUC
- hello.C
- hello.c
- hello.f
- hello.pas

ファイルのコピー

□ コマンド

% cp [オプション] ファイル名1 ファイル名2

% cp ファイル名1 ファイル名2

% cp -i ファイル名1 ファイル名2

% cp -i ファイル名1 ディレクトリ名

演習 16 先ほど存在を確認したファイルを他のファイルにコピーしなさい。

例 % cp -i hello.C bonjour.C

[注意]

cp を用いるときは、必ず -i というオプションをつける!

演習 17 cp を用いるときには、なぜ-i オプションをつけることが望ましいのだろうか。理由は？以下を行なうとどうなるか？

例 % cp -i hello.pas bonjour.C

ファイルの移動

□ ファイルの移動, ファイル名の変更 mv (move)

□ コマンド

```
% mv [オプション] ファイル名1 ファイル名2
```

```
% mv ファイル名1 ファイル名2
```

```
% mv -i ファイル名1 ファイル名2
```

```
% mv -i ファイル名1 ファイル名2 ディレクトリ名
```

演習 18 hello.c を tmp.c というファイル名にしてください。

演習 19 apr にある hello.c を may に移動してください。

演習 20 apr にあるファイルを他のディレクトリに移動してください。

演習 21 mv と cp の違いは？

演習 22 以下により何がおきるか？

```
% mv -i hello.? ../jul
```

ワイルドカード

□ ? → 任意の一文字

hello.f, hello.c, hello.C ⇒ hello.?

□ * → 任意の長さの文字

hello.f, hello.c, hello.C, hello.pas ⇒ hello.*

演習 23 jun ディレクトリにおいて，以下を実行せよ．

```
% mksamp cmp
```

演習 24 jun ディレクトリ内にある全ファイルを，may ディレクトリに移動せよ．ワイルドカードを用いる．

[注意]

ワイルドカードは，どんなコマンドにも使える．

ファイルの消去

- ファイルの消去 rm (remove)
- コマンド
 - % rm [オプション] ファイル名 1
 - % rm ファイル名 1
 - % rm -i ファイル名 1

演習 25 tmp.c を hello.c にコピーし , tmp.c を消去しなさい .

演習 26 rm と mv の違いは ?

演習 27 以下により何がおきるか ?

```
% rm -i hello.pas hello.c
```

[注意]

rm や mv を用いるときも , 必ず -i というオプションをつける!

ファイル内容の表示

□ コマンド (cat, more, less)

```
% cat [オプション] file1 file2 file3
```

```
% more [オプション] file1 file2 file3
```

```
% less [オプション] file1 file2 file3
```

演習 28 以下のファイルの中味を表示しなさい。

- text.EUC
- hello.C
- hello.c
- hello.f
- hello.pas

その他のコマンド

使い方を覚えると便利だと思われるコマンド

- head, tail
- find, grep, egrep, fgrep
- which
- sort, wc
- cal, date, clear
- chmod
- tar, lha
- compress, uncompress, gzip, gunzip
- nkf, banner
- lpr, a2ps, lpq, lpstat, lprm, cancel

演習 29 上記のコマンドをコンピュータ利用の手引き 2006 などを用いて調べなさい。