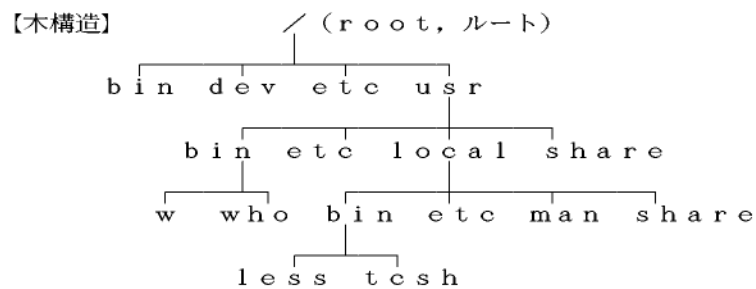


3章「ファイルシステム」

中島康彦

§3. 1 UNIXファイルシステムの特徴

ファイルシステム (ファイルの管理体系)



▶ 通常ファイル

`/usr/bin/who`

▶ ディレクトリファイル

`/usr/local/bin`

▶ 特殊ファイル(入出力装置や記憶装置等)

`/dev/pts/*` ... ターミナル

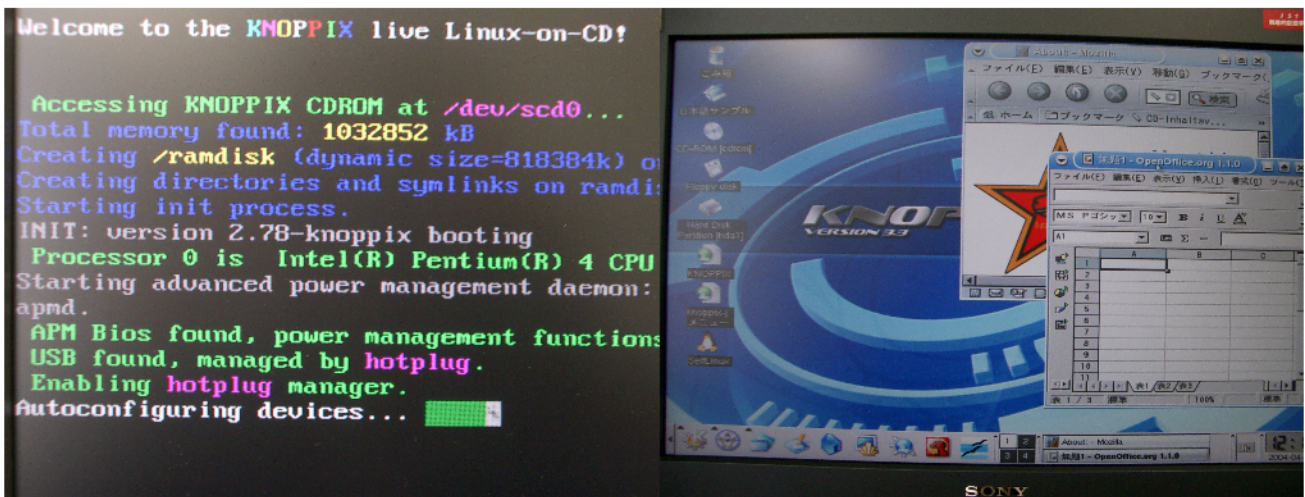
`/dev/rd/*` ... ハードディスク

`/dev/mem` ... 主記憶

§3. 2 1CD-UNIX

「ファイルシステム=ハードディスク」ではない

- ▶ CDROMやメモリ上にファイルシステムを作れる
- ▶ OSのインストールが不要
- ▶ ウイルス感染/勝手な設定変更の心配がない
- ▶ 個人のデータはUSBメモリに格納
- ▶ パソコンをゲーム専用機のように使える



§3. 3 カレントワーキングディレクトリ

▶ pwd

カレントワーキングディレクトリ(CWD)の表示

▶ cd

ホームディレクトリ(ログイン直後のディレクトリ)へ

相対パス (CWDからの相対的順路) による移動

▶ cd . ⇒ CWDへ(=移動しない)

▶ cd ./bin ⇒ CWDの下のbinへ

▶ cd bin ⇒ CWDの下のbinへ

▶ cd .. ⇒ 1つ上へ

▶ cd ../local ⇒ 1つ上に戻りlocalへ

▶ cd ../../local ⇒ 2つ上に戻りlocalへ

絶対パス (ルートからの順路) による移動

▶ cd /usr/local/bin

§3. 4 ディレクトリスタック

ディレクトリスタックを使った移動

`pushd, popd, dirs`

1. % `cd /usr` ⇒ `/usr`へ移動
2. % `dirs` ⇒ `/usr` (ディレクトリスタックの表示)
3. % `pushd /tmp` ⇒ CWDをスタックに積み/`tmp`へ
4. % `dirs` ⇒ `/tmp /usr`
5. % `pushd /usr/local` ⇒ CWDをスタックに積み/`usr/local`へ
6. % `dirs` ⇒ `/usr/local /tmp /usr`
7. % `popd` ⇒ `/tmp`に戻る
8. % `dirs` ⇒ `/tmp /usr`
9. % `popd` ⇒ `/usr`に戻る
10. % `dirs` ⇒ `/usr`
11. % `pwd` ⇒ `/usr`に戻っている
12. % `popd` ⇒ `Directory stack empty.` (スタックが空)

§3. 5 ディスク使用量

ディスク使用量の表示

`du, xdu`

- ▶ `du -s`
CWD以下のディスク総使用量 (単位は1KB)
- ▶ `du`
CWD以下のディレクトリ毎のディスク使用量
- ▶ `du -s ディレクトリ名`
- ▶ `du ディレクトリ名`
特定のディレクトリに関する表示
- ▶ `xdu ディレクトリ名`
X Window System上でグラフ表示

§3. 6 ディレクトリの操作

ディレクトリの作成と複写

`mkdir, mkdirhier, cp`

- | | | |
|----|--------------------------------|---------------------------|
| | <code>% cd</code> | ⇒ ホームディレクトリへ移動 |
| 1. | <code>% mkdir chap03</code> | ⇒ ディレクトリchap03を作成 |
| | <code>% cd chap03</code> | ⇒ ディレクトリchap03へ移動 |
| 2. | <code>% mkdir abc</code> | ⇒ ディレクトリabcを作成 |
| | <code>% ls</code> | ⇒ 一覧表示 (abcができています) |
| 3. | <code>% cp -r abc def</code> | ⇒ ディレクトリabcのコピーdefを作成 |
| | <code>% ls</code> | ⇒ 一覧表示 (defができています) |
| 4. | <code>% mkdirhier a/b/c</code> | ⇒ 深いディレクトリを作成 |
| | <code>% ls</code> | ⇒ 一覧表示 (aができています) |
| | <code>% ls -R a</code> | ⇒ 一覧表示 (a/bとa/b/cもできています) |
| 5. | <code>% cp -r a d</code> | ⇒ ディレクトリa全体のコピーdを作成 |
| | <code>% ls</code> | ⇒ 一覧表示 (dができています) |
| | <code>% ls -R d</code> | ⇒ 一覧表示 (d/bとd/b/cもできています) |

§3. 6 ディレクトリの操作(続き)

ディレクトリの移動と削除

`mv, rmdir, rm`

- | | | |
|----|------------------------------|----------------------------|
| 1. | <code>% mv def ghi</code> | ⇒ ディレクトリdefをghiに移動 (名前の変更) |
| | <code>% ls</code> | ⇒ 一覧表示 (defが消えghiができた) |
| 2. | <code>% mv ghi abc</code> | ⇒ ディレクトリghiをディレクトリabcの中に移動 |
| | <code>% ls</code> | ⇒ 一覧表示 (ghiが消えた) |
| | <code>% ls abc</code> | ⇒ ディレクトリabcの一覧表示 (ghiができた) |
| 3. | <code>% rmdir abc/ghi</code> | ⇒ ディレクトリghiの削除 |
| 4. | <code>% rmdir abc</code> | ⇒ ディレクトリabcの削除 |
| | <code>% ls</code> | ⇒ 一覧表示 (abcが消えた) |
| 5. | <code>% rm -ri a</code> | ⇒ 確認にyと答えディレクトリa全体を削除 |
| 6. | <code>% rm -rf d</code> | ⇒ 確認なしにディレクトリd全体を削除 (危険) |
| | <code>% ls</code> | ⇒ 一覧表示 (aとdが消えた) |
-

§3. 7 ファイルの操作

ファイルの作成と複写

cat, cp

1. % **cat > abc**
01234 [Enter]
56789 [Enter]
[Ctrl]+d ⇒ ファイルabcを作成しキー入力を書き込む
% **ls** ⇒ 一覧表示 (abcができています)
2. % **cp abc def** ⇒ ファイルabcの内容をファイルdefに複写
% **ls** ⇒ 一覧表示 (defができています)
3. % **cat def** ⇒ ファイルdefの内容を表示

§3. 7 ファイルの操作(続き)

ファイルの移動と削除

mv, rm

1. % **mkdir tmp** ⇒ ディレクトリtmpを作成
1. % **mv def ghi** ⇒ ファイルdefをghiに移動 (名前の変更)
% **ls** ⇒ 一覧表示 (defが消えghiができた)
2. % **mv ghi tmp** ⇒ ファイルghiをディレクトリtmpの中に移動
% **ls** ⇒ 一覧表示 (ghiが消えた)
% **ls tmp** ⇒ ディレクトリtmpの一覧表示 (ghiができた)
3. % **rm -i tmp/ghi** ⇒ 確認にyと答えファイルghiを削除
4. % **rm -f abc** ⇒ 確認なしにファイルabcを削除 (危険)
% **ls** ⇒ 一覧表示 (abcが消えた)

§3. 8 リンク

ハードリンクの作成と削除 `ln`

1. `% cat > abc`
`01234 [Enter]`
`[Ctrl]+d` ⇒ `abc`を作成し01234を書き込む
2. `% ln abc def` ⇒ `abc`に対するハードリンク`def`を作成
3. `% cat def` ⇒ `def`の内容も01234
4. `% cat > abc`
`11111 [Enter]`
`[Ctrl]+d` ⇒ `abc`に11111を上書きする
5. `% cat def` ⇒ `def`の内容も11111に変わっている
6. `% rm abc` ⇒ `abc`を削除 (ハードリンクが切れる)
7. `% cat def` ⇒ `def`の内容は11111のまま
8. `% cat > abc`
`22222 [Enter]`
`[Ctrl]+d` ⇒ `abc`を作成し22222を書き込む
9. `% cat def` ⇒ `def`の内容は11111のまま
10. `% rm def` ⇒ リンク`def`を削除

§3. 8 リンク(続き)

シンボリックリンクの作成と削除 `ln -s`

1. `% cat > abc`
`01234 [Enter]`
`[Ctrl]+d` ⇒ `abc`を作成し01234を書き込む
2. `% ln -s abc def` ⇒ `abc`に対するシンボリックリンク`def`を作成
3. `% cat def` ⇒ `def`の内容も01234
4. `% cat > abc`
`11111 [Enter]`
`[Ctrl]+d` ⇒ `abc`に11111を上書きする
5. `% cat def` ⇒ `def`の内容も11111に変わっている
6. `% rm abc` ⇒ `abc`を削除
7. `% cat def` ⇒ 「ファイルはありません」★★★★
8. `% cat > abc`
`22222 [Enter]`
`[Ctrl]+d` ⇒ `abc`を作成し22222を書き込む
9. `% cat def` ⇒ `def`の内容が22222に変わっている★★★★
10. `% rm def` ⇒ リンク`def`を削除

§3. 9 ファイル情報の表示

ファイル情報の表示 `file`, `ls`

▶ `file` ファイル

種別の表示

▶ `ls` ファイル

属性の表示 (ファイル省略時はCWDを仮定)

オプション(複数同時指定可能)

- l ⇒ 1行に1ファイルを表示
- d ⇒ ディレクトリの場合, 中まで表示しない
- t ⇒ 変更時刻順 (新しい順) で表示
- r ⇒ 表示順の反転
- F ⇒ ファイル名の後ろに種別を記号表示
 - / ... ディレクトリ
 - * ... 実行可能なファイル
 - @ ... シンボリックリンク

§3. 9 ファイル情報の表示(続き)

-l ⇒ ファイルモード, ハードリンク数, 所有者, 所属グループ, サイズ, 最終変更時刻

```
+ (d)ディレクトリ (l)シンボリックリンク (-)通常ファイル
|+-----user---r- 所有者による読み取り
||+-----user---w- 所有者による書き込み
|||+-----user---x- 所有者による実行/ディレクトリ検索
||||+-----group--r- グループによる読み取り
|||||+-----group--w- グループによる書き込み
||||||+-----group--x- グループによる実行/ディレクトリ検索
|||||||+---others-r- その他のユーザによる読み取り
|||||||+---others-w- その他のユーザによる書き込み
|||||||+---others-x- その他のユーザによる実行/ディレクトリ検索
|||||||
-lrwxlrwxlrwx
drwxr-xr-x 2 ynakashi faculty 1024 1月29日 20:06 Bin/
drwxr-xr-x 5 ynakashi faculty 1024 1月23日 15:44 Lib/
drwxr-xr-x 2 ynakashi faculty 1024 1月25日 14:15 Mail/
lrwxr-xr-x 1 ynakashi faculty 5 2月 5日 21:36 sample@ -> setup
-rwxr-xr-x 1 ynakashi faculty 175 1月29日 17:04 setup*
```

§3. 10 ファイル属性の変更

ファイル属性の変更 touch, chmod

- ▶ touch ファイル
最終変更時刻の変更
オプション省略時は現在時刻に設定

- ▶ chmod モード ファイル
ファイルモードの変更
user/group/others/all毎にrwxを指定
 <u><g><o> <a>
 -rwxrwxrwx
 --- (0) --x (1) -w- (2) -wx (3)
 r-- (4) r-x (5) rw- (6) rwx (7)

§3. 10 ファイル属性の変更(続き)

- ▶ chmod go-w
 - ▶ chmod 755
 -rwxr-xr-x ... ユーザ以外の書き込みを禁止
 - ▶ chmod go-rwx
 - ▶ chmod 700
 -rwx----- ... 読み/書き/実行/検索を禁止
 - ▶ chmod go+rx
 - ▶ chmod 755
 -rwxr-xr-x ... 読み込み/実行/検索を許可
 - ▶ chmod go+w
 - ▶ chmod 777
 -rwxrwxrwx ... 書き込みを許可
 - ▶ chmod a-rwx
 - ▶ chmod 000
 ----- ... 全て禁止
-

§3. 11 ファイルの検索

ファイルの検索 `which`, `find`

▶ `which` コマンド

コマンドの存在場所を表示

▶ `find`

ファイルを検索

オプション(複数同時指定可能)

ディレクトリ ⇒ 検索開始位置の指定

`-name` 名前 ⇒ 名前を指定

`-user` ユーザ名 ⇒ 所有者を指定

`-group` グループ ⇒ グループを指定

`-size` ±サイズ ⇒ サイズを指定

`-mtime` ±日数 ⇒ 最終変更時刻からの日数

`-type` タイプ ⇒ タイプを指定

`f` ⇒ 通常ファイル

`d` ⇒ ディレクトリ

`l` ⇒ シンボリックリンク

§3. 11 ファイルの検索(続き)

▶ `find . -name "M*"`

CWD以下のMで始まるファイルを検索

▶ `find . -size +100`

▶ `find . -size +100 -exec ls -dl {} \;`

サイズが100Kバイト以上のファイルを検索

▶ `find . -mtime -1`

▶ `find . -mtime -1 -exec ls -dl {} \;`

過去24時間以内に変更されたファイルを検索

▶ `find /bin -type l -exec ls -dl {} \;`

シンボリックリンクを検索

§3. 12 例題

各ホームディレクトリ以下の, 過去30日以内に變更されたファイルを検索せよ.

▶ `find ~ -mtime -30`

§3. 13 今日の課題

あなたが管理するWEBサーバプログラム (httpd) にセキュリティホールが発見された. 大至急, プログラムを入れ換えなければ, あなたのサーバは壊されてしまう. 新しいプログラム(httpd)を入手したとして, httpdの場所を探し, 入れ換える手順について, 今日紹介したコマンド列を用いて具体的に説明せよ.

宛先: nakashim@econ.kyoto-u.ac.jp

件名: unix1-学生番号

名前(忘れずに)

- ▶ コピーしたい範囲を左ドラッグで選択し,
- ▶ 次にメール作成画面で中クリック.

今日はここまで