情報処理論1a

4章「標準入出力/パイプ/フィルタ」

中島康彦

§4. 0 今日の作業ディレクトリを作る

1. % `cd` ⇒ ホームディレクトリへ移動
2. % `mkdir chap04` ⇒ ディレクトリchap04を作成
3. % `cd chap04` ⇒ ディレクトリchap04へ移動
4. % `pwd` ⇒ 作業ディレクトリの確認

§4. 1 標準入出力

多くのプロセスが以下の入出力方法を採用

- 入力ファイルが省略された場合，標準入力を使用
  キー入力またはファイルを接続

	標準入力	標準出力	標準エラー出力
	↓	↓	↓
`cat file1`	なし	画面	画面
`cat`	キー入力	画面	画面
`cat < file1`	`file1`	画面	画面

§4. 1 標準入出力(続き)

- 実行結果は標準出力，エラーは標準エラー出力へ
  画面またはファイルへ接続
  `>`,`>&`は上書き．`>>`,`>>&`は追加

	標準入力	標準出力	標準エラー出力
	↓	↓	↓
`cat file1 > file2`	なし	`file2`	画面
`cat > file2`	キー入力	`file2`	画面
`cat < file1 > file2`	`file1`	`file2`	画面
`cat file1 >& file3`	なし	`file3`	`file3`
`cat >& file3`	キー入力	`file3`	`file3`
`cat < file1 >& file3`	`file1`	`file3`	`file3`

§4. 2 ファイル表示

- less ファイル(省略時は標準入力から)

h	⇒	lessのマニュアル表示
j ↓ [Enter]	⇒	1行進む
k ↑	⇒	1行戻る
f [Space]	⇒	1ページ進む
b	⇒	1ページ戻る
d	⇒	半ページ進む
u	⇒	半ページ戻る
/xxx	⇒	xxxの前方検索
?xxx	⇒	xxxの後方検索
n	⇒	検索を繰り返す
N	⇒	逆方向へ検索を繰り返す
q	⇒	lessの終了

- more ファイル(省略時は標準入力から)
  lessが無い場合に使用する. 一般に逆方向スクロールはできない.

§4. 2 ファイル表示(続き)

- head -行数 ファイル(省略時は標準入力)
  先頭から行数分を表示(省略時は10)

- tail -行数 ファイル(省略時は標準入力)
  末尾から行数分を表示(省略時は10)

- tail +行数 ファイル(省略時は標準入力)
  先頭から「行数-1」分を非表示

- xd -c ファイル(省略時は標準入力)
  数値ダンプおよび文字により表示

§4. 3 パイプ

パイプ … プロセスの標準出力を他プロセスの標準入力へ接続する機構

	標準入力	標準出力	標準エラー出力
	↓	↓	↓
`cat file1 | less`	なし	lessへ	画面
`cat | less`	キー入力	lessへ	画面
`cat < file1 | less`	file1	lessへ	画面
`cat file1 |& less`	なし	lessへ	lessへ
`cat |& less`	キー入力	lessへ	lessへ
`cat < file1 |& less`	file1	lessへ	lessへ

§4. 3 パイプ(続き)

1. `% cat > abc`
   `01234[Enter]`
   `56789[Enter]`
   `[Ctrl]+d` ⇒ abcを作成する
2. `% cat > def`
   `abcde[Enter]`
   `[Ctrl]+d` ⇒ defを作成する
3. `% cat -n abc def` ⇒ 出力に行番号が付く
   ```
    1  01234
    2  56789
    3  abcde
   ```
4. `% cat -n abc def | cat -n` ⇒ さらに行番号が付く
   ```
    1  1  01234
    2  2  56789
    3  3  abcde
   ```
5. `% cat -n abc def | tee ghi | cat -n` ⇒ パイプの途中をファイルghiへ出力
   ```
    1  1  01234
    2  2  56789
    3  3  abcde
   ```
6. `% cat ghi` ⇒ 3.と同じ内容が入っている
   ```
    1  01234
    2  56789
    3  abcde
   ```

§4. 4 単機能フィルタ

- expand ファイル(省略時は標準入力)
  タブ(Tab) → 複数の空白に変換

  aaa△　bbbbb△cc　… 12文字
  ↓
  aaa＿＿＿bbbbb＿＿cc　… 18文字

- unexpand -a ファイル(省略時は標準入力)
  複数の空白 → タブ(Tab)を使って圧縮

  aaa＿＿＿bbbbb＿＿cc　… 18文字
  ↓
  aaa△　bbbbb△cc　… 12文字

§4. 4 単機能フィルタ(続き)

- cut -cリスト ファイル(省略時は標準入力)
  各行の指定位置の文字を表示
  -c1,3,5,7 ⇒ 行頭から1,3,5,7文字目
  -c1-3,5-7 ⇒ 行頭から1～3と5～7文字目

- cut -dデリミタ -fリスト ファイル(〃)
  各行の指定フィールドを表示
  -dデリミタ省略時はTabを仮定
  -d':' -f1,3 ⇒ :で区切った1,3番フィールド

- cat /etc/passwd
- cat /etc/passwd | cut -d':' -f1,6
  ユーザ名とホームディレクトリのみ表示

§4. 4 単機能フィルタ(続き)

- `paste -d`デリミタリスト ファイル(- は標準入力)
  複数ファイルの内容を同じ行ごとに列方向(右方向)に並べる.

- `paste -d' ' a b c`
  aの1行目 bの1行目 cの1行目
  aの2行目 bの2行目 cの2行目
  aの3行目 bの3行目 cの3行目

- `paste -d':;/' a - -`
  aの1行目:標準入力の1行目;標準入力の2行目
  aの2行目:標準入力の3行目;標準入力の4行目
  aの3行目:標準入力の5行目;標準入力の6行目

§4. 5 文字置換フィルタ

- `tr -d` 文字 (常に標準入力)
  文字を削除する.

`\n`	⇒	改行
`\t`	⇒	Tab
`[a-z]`	⇒	a〜zまでの文字
`[a-zA-Z0-9]`	⇒	a〜z,A〜Z,0〜9まで
`[:alpha:]`	⇒	英字...`[a-zA-Z]`
`[:digit:]`	⇒	数字...`[0-9]`
`[:alnum:]`	⇒	英数字...`[a-zA-Z0-9]`
`[:xdigit:]`	⇒	16進文字...`[a-fA-F0-9]`
`[:space:]`	⇒	空白,Tab,改行
`[:lower:]`	⇒	英小文字...`[a-z]`
`[:upper:]`	⇒	英大文字...`[A-Z]`

- `tr -d '[a-z]' < /etc/passwd`
  英小文字を削除

§4. 5 文字置換フィルタ(続き)

- `tr 文字1 文字2 (常に標準入力)`
  文字1を文字2に置き換える.

- `tr ':' '\n' < /etc/passwd`
  :を改行に置き換える.

- `tr ':' '\t' < /etc/passwd`
  :をTabに置き換える.

- `tr '[0-9]' 'X' < /etc/passwd`
  数字をXに置き換える.

- `tr '[a-z]' '[A-Z]' < /etc/passwd`
  英小文字を英大文字に置き換える.

§4. 6 文字列置換フィルタ

正規表現(Basic Regular Expression)

^ ⇒ 行頭　　$ ⇒ 行末
. ⇒ 任意の1文字　　* ⇒ 0回以上
[] ⇒ 文字クラス　　\ ⇒ 次文字を通常文字と解釈
通常文字 ⇒ その文字自身

```
abc             ⇒ abcに一致
^abc            ⇒ 行頭のabc
abc$            ⇒ 行末のabc
^abc$           ⇒ 一行がabcのみ
abc.1           ⇒ abc01, abcA1, abcJ1
abc.*1          ⇒ abc1, abc01, abc001, abc0001
abc[2-9]1       ⇒ abc21, abc31
abc[2-9A-Z]1    ⇒ abc21, abc31, abcA1
abc[^2-9A-Z]1   ⇒ abc01, abc11, abca1
```

§4. 6 文字列置換フィルタ(続き)

- `sed 's/文字列1/文字列2/' ファイル`
- `sed 's/文字列1/文字列2/g' ファイル`

  上段:各行の最初の文字列1を文字列2に置換

  下段:全ての文字列1を文字列2に置換

- `sed 's/[0-9][0-9]*/xxx/' /etc/passwd`

  各行の最初の数値を**xxx**に置き換える.

- `sed 's/[0-9][0-9]*/xxx/g' /etc/passwd`

  全ての数値を**xxx**に置き換える.

- `sed 's/^[a-z0-9]*/<&>/' /etc/passwd`

  ユーザ名を<ユーザ名>に置き換える.

§4. 7 より高度なフィルタ

- `awk 'BEGIN{FS="フィールドセパレータ"}`
  `/文字列1/{print フィールド1}`
  `/文字列2/{print フィールド3} … ' ファイル`

  一致する行毎に指定した形式で表示

  `$0` ⇒ 行全体　`$1`以降 ⇒ フィールド位置指定

- `awk 'BEGIN{FS=":"}/^root:/{print $3}'`
  `/etc/passwd`

  ユーザ名rootのUID番号を表示する.

- `awk 'BEGIN{FS=":"}/^[a-m]/{print $1}`
  `/^r/{print $0}' /etc/passwd`

  a～mで始まるユーザ名, および, rで始まるユーザの行全体を表示する.

§4. 8 日本語コード変換フィルタ

- `nkf -j` ファイル(省略時は標準入力)
  `ISO-2022-JP(7-bit JIS)`コードに変換する. 主にメールで用いられる日本語コード体系

- `nkf -s` ファイル(省略時は標準入力)
  `MS`漢字(シフト`JIS`)コードに変換する. 主に`Microsoft`製パソコンソフトで用いられる体系

- `nkf -e` ファイル(省略時は標準入力)
  `EUC(AT&T)`コードに変換. `Extended Unix Code`の名前の通り`UNIX`で用いられる体系

§4. 9 テキストファイル以外の無数の変換ツール

- `WAV,MP3,MIDI,...` 音声,音楽関係
- `AVI,MPEG,...` 映像関係
- `JPEG,GIF,TIFF,BMP,...` 画像関係
- `TeX,Postscript,PDF,...` 文書関係
- その他

多くのフリーソフトウェアが存在.
ただし, 著作物の取り扱いにはくれぐれも注意
- 変換できるからと言って, 何でも許されるわけではない
- 著作者の立場で考えて, されて嫌と思うようなことはしない

§4. 10 例題

ls -l の出力から，ファイルモード，サイズ，ファイル名を切り出すコマンドを考えよ．なお，下記の例は除く．

```
ls -l | sed "s/__*/;/g" | cut -d';' -f1,5,9
```

- `ls -l | awk 'BEGIN{FS=" "}//{print $1,$5,$9}'`

§4. 11 今日の課題

calコマンドにより，当月のカレンダが出力されることは以前に説明した．1行に「1 2 . . . 30 31」とだけ出力するには，どうすればよいか．以下の「...」の部分を考えよ．

- `cal | ...`

宛先: nakashim@econ.kyoto-u.ac.jp
件名: unix1-学生番号

名前(忘れずに)

今日はここまで