

awk プログラミング

*パターンによる行指定：パターンにマッチした行を抜き出し、アクションを実行する。

- 例 1) `NR > 1 { アクション }` 行数が 1 を超えていればアクションを実行
例 2) `$4 ~ /\.0/ { アクション }` 第 4 フィールドが任意の文字 + 0 であれば、
アクションを実行

[BEGIN と END は特殊、パターンを省略するとすべての行が対象となる]

- BEGIN awk が行読み込みを始める前に実行するアクションを指定
- END awk がすべての行読み込みを終えた後に実行するアクションを指定

[日本語の入力]

vi エディタで挿入モードに入り、`shift キー + スペースバー` で日本語変換モードに入る。
ローマ字で入力し、`スペースバー` で変換、`リターンキー` で確定する。(こまめに確定した方がよい)
日本語入力モードから抜けるのも `shift キー + スペースバー` で行う。

課題 1

- 1) ssh コマンドで Linux サーバにログインする。
- 2) vi `ls.awk` により、下に示すような awk プログラムを作成する。
- 3) `ll | awk -f ls.awk` を実行する。

ls.awk

```
#初期化
BEGIN {
    size = 0 #バイト数を格納する変数の初期化
}

#total ブロック表示を除く各行について
NR > 1 {
    printf "%3d %s\n", NR-1, $0; #行番号を追加
    size += $5
}

#最後にファイル総数とバイト数とを表示する
END {
    print "ファイル総数="NR-1 " 総バイト数="size"バイト"
```

課題 2

- 1) vi `kamoku.dat` により、裏面に示すようなデータを作成する。
- 2) vi `hisshu.awk` により、裏面に示すような awk プログラムを作成する。
- 3) vi `sentaku.awk` を実行し、`:r hisshu.awk` により読み込んだ awk プログラムを編集して、裏面に示すような awk プログラムを作成する。

- 4) vi tsunen.awk を実行し, :r sentaku.awk により読み込んだ awk プログラムを編集して, 裏面に示すような awk プログラムを作成する。
- 5) awk -f hisshu.awk kamoku.dat を実行する。
- 6) awk -f sentaku.awk kamoku.dat を実行する。
- 7) awk -f tsunen.awk kamoku.dat を実行する。

kamoku.dat

講読と表現 1	2	required	11
英語 1	2	required	11
情報科学	2	required	11
プログラミング論 1	2	optional	11
プログラミング論 2	2	optional	12
ゼミナール 1	2	required	10
ネットワークシステム	2	optional	21
プログラミング言語演習	1	optional	21
UNIX 演習	1	optional	22
ゼミナール 2	2	optional	20

(第 2 フィールドは単位数, 第 4 フィールドは十の位が年次, 一の位が 1:前期, 2:後期, 0:通年)

hisshu.awk

```
$3=="required" { sum += $2; printf "%-24s%2d¥n", $1, $2}
END { printf "%-24s%2d¥n", "単位数合計", sum}
```

sentaku.awk

```
$3=="optional" { sum += $2; printf "%-24s%2d¥n", $1, $2}
END { printf "%-24s%2d¥n", "単位数合計", sum}
```

tsunen.awk

```
$4 ~ /\.0/ { sum += $2; printf "%-24s%2d¥n", $1, $2}
END { printf "%-24s%2d¥n", "単位数合計", sum}
```