情報工学演習1
第3回

Linuxの使用法2 (パイプ，プロセス管理)

今回の目標

- 前回のLinuxコマンドの続き
- Linuxコマンドについて理解を深めよう.
- 本演習は前回と同じ環境**(Replit の C のモード)**で行ってください.

コマンド入力を速くするコツ
(1) キーボードの上下方向キーで
コマンド入力履歴を辿れる.
(2) コマンドやファイル名など,
途中まで入力したところで Tabキーを
押すと続きが補完される.

基本コマンド：ファイルの閲覧など

コマンド名	説明
less	more コマンドの対義語 テキストファイルの閲覧
wc	***w**ord **c**ount* ファイルの行数などの情報を表示する
echo	引数の内容を出力する ワイルドカードは，展開して出力される
grep	***g**lobal **r**egular **e**xpression **p**rint* ファイル内の文字列を検索して表示する
egrep	***e**xtended **grep*** grepよりも高度な正規表現を使える 正規表現を使うとき検索文字列を””で囲む

演習室では使えたが
Replitでは使えなかった.
コマンド有無は環境依存

less コマンド

環境によっては
こんなコマンドも
ありますよと参考程度

- less 【ファイル名】
- ファイル閲覧中の操作
  - SPACE ： 1画面分進む
  - j ： 1行進む
  - k ： 1行戻る
  - q ： ファイルの閲覧を終了する
  - / ： ファイル内の前方検索
  - ? ： ファイル内の後方検索

wc コマンド

- wc 【ファイル名】
- オプション
  - -l ： 行数を表示
  - -c ： バイト数を表示
  - -m ： 文字数を表示
  - -w ： 単語数を表示
  - -L ： 最も長い行の長さを表示

echo コマンド

- echo 【任意】
- 使用例
  - `echo Hello world`
    「Hello world」がそのまま表示される.
  - `echo *.c`
    「main.c rand.c」のように作業ディレクトリ内でワイルドカードにマッチするものがすべて表示される.

grep コマンド

- grep 【検索したい文字列】【ファイル名】
  - 使用例
  - `grep int *.c`
    作業ディレクトリ内の拡張子が .c であるすべての
    ファイルについて，int という文字列を検索する．
- オプション
  - -G ： 検索文字列に基本的な**正規表現**を使う
  - -E：拡張正規表現を使う
  - -F ： 固定文字列を使った検索を行う
  - -i ： 検索文字列の大文字小文字を区別しない

正規表現は様々な
文字列パターンを
表現し，
マッチするものを
すべて検索できる

grep，オプション付きgrep，egrepで
使用できる正規表現が異なる場合があるので注意

(拡張)正規表現

参考：正規表現メモ
http://www.kt.rim.or.jp/~kbk/regex/regex.html

正規表現	
.	改行文字以外の任意の1文字に一致
*	直前の１文字の０回以上の繰り返しに一致
^	行の先頭を表す
$	行の末尾を表す
[ ]	括弧内の任意の文字に一致. 「-」で範囲指定可能 括弧内の最初に ^ があると not の意味になる
+	直前の文字の１個以上の連続
?	直前の文字の０または１文字に一致
{n}	直前の文字 n 個に一致
{n,m}	直前の文字 n 個以上m個以下に一致
{n,}	直前の文字 n 個以上に一致
{,m}	直前の文字 m 個以下に一致
p1|p2	p1 または p2 のいずれかに一致
(pattern)	pattern をグループ化する

正規表現の例

- **ワイルドカードとは別物**であることに注意する.
- **grep で動かなければ egrep を試す.**
- `3.5`
  3a5, 3b5, 3/5, などにマッチ.
- `a1*`
  a, a1, a11, a111, a1111, a11111, … にマッチ.
  繰り返し0回でもマッチするので注意.
- `[1-59]`
  1, 2, 3, 4, 5, 9 に一致.
  つまり, 「1～5」 と「9」.
  「1」から「59」までではない.
- `[0-9]{4}`
  4桁の数字にマッチ.

リダイレクト

- 「>」記号によって，あるコマンドの出力を，ファイルとして保存できる.
- 使用例
  - `ls -l > result.txt`
    ls -l の結果を result.txt というファイルに保存
  - `grep int main.c > int_main.txt`
    main.c 内の int を含む行を検索して，
    int_main.txtというファイルに保存
- リダイレクト先としてデバイスを指定することもできる
  - `ls -l > /dev/null`
    ls -l の結果を読み捨てる.
    /dev/null は nullデバイスと呼ばれる.

追加書き込み

- 「>>」と2つ続けると，あるコマンドの出力を，
  ファイルの末尾に追加書き込みできる.
- `command > file` とすると，
  fileの内容がcommandの出力で**上書き**される.
  上書き保存のため，前に書き込んだ内容が消えてしまう.
- `command >> file`
  command の標準出力を，fileの**末尾に追加**する.
- 例：

  ```
  echo ----- begin ----- >> file
  ```

  任意の文字列をファイルに追加する.

パイプ

- 「|」(縦線，パイプ)によって，コマンドの出力を，
  別のコマンドの入力として渡す.
- `command1 | command2`
- パイプの使用例
- a.out は前回作成した乱数生成プログラムとする.
- `./a.out | wc -l`
  a.out の出力の行数を表示する.
  前回の課題をコンパイルしたものなら，100のはず.
- `./a.out | grep ^[0-9]$`
  生成された100個の[1,100]の一様乱数のうち，
  1桁のものを表示する.
- `./a.out | grep ^[0-9]$ | wc -l`
  生成された100個の[1,100]の一様乱数のうち，
  1桁のものの数を表示する.

バックグラウンドプロセス

- `command &`
  command をバックグラウンドで実行する.
- jobs コマンド
  バックグラウンドで実行中のプロセスを確認する.
- kill コマンド
  プロセスを停止させる.
  jobs でプロセス番号を確認し,
  「`kill %1`」のように指定する
  - 「`kill 1`」だと, osの動作上重要なプロセスを停止させて osをハングアップさせる可能性があるので注意

プログラムの停止

- プログラミングの結果，無限ループに入ってしまうなどして処理がいつまでも終わらないことがある.
- Ctrl+c でプログラムを停止できる.
- Ctrl+z だとプログラムはサスペンドされており，一時停止されたバックグラウンドプロセスとして残っている（＝メモリを占有したまま）.
- サスペンドしたプロセスは，`jobs`コマンドから確認して`kill`できる.
- `fg` コマンドでサスペンドを解除してフォアグラウンドに戻すこともできる.

補足：改行コードに注意しよう

- 改行コードの違いによって，見た目にはわからないが
  プログラムの動作が変化する場合があるので注意.

改行コード	記号	cat上の記号	OS
LF	¥n	$	UNIX, Linux, Mac OSのX以降
CRLF	¥r¥n	^M$	Windows
CR	¥r	^M	Mac OS 9以前

- 改行コードの確認： `cat -e sample.txt`

```
cat -e sample_linux.txt
1$
2$
3$
```

```
cat -e sample_windows.txt
1^M$
2^M$
3^M$
```

例年はWindows併用のため問題が生じる場合があった.
今回は Replit上のLinux使用のため改行コードLFで問題ないはず.

第3回課題

- 自分の学籍番号の下4桁を乱数のシードとしたときの，前回の課題で得られる100個の乱数の度数分布を，Linuxコマンドによって作成する．
  - 「1~9」，「10~19」，「20~29」，…，「90~99」，「100」という11個のカテゴリについて出現個数を数えた結果を示せ．

(1) まず，上記の100個の乱数を1行に1つずつ保存したテキストファイルを生成するコマンドを実行する．
(2) 次に，先程生成したテキストファイルをもとに，カテゴリごとに度数を表示するコマンドを実行する．

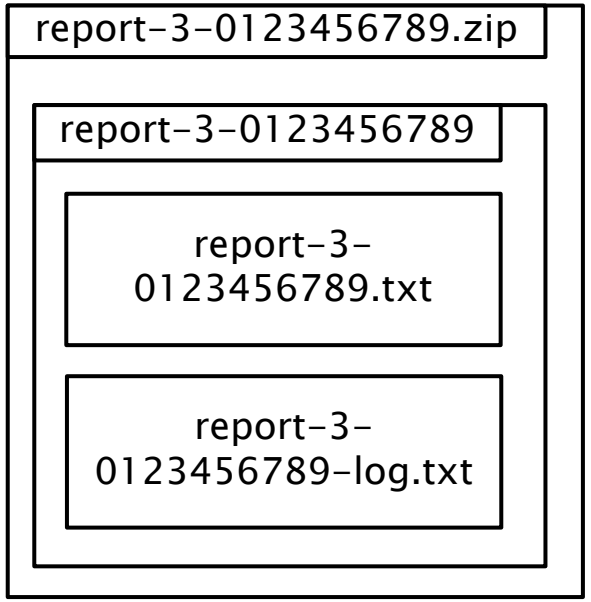
- 目で見て数えたり度数分布を生成するプログラムを組むのではなく，**Linuxコマンドを使って数える**こと．

第3回課題の提出物

- report-3-学籍番号.txt
  - 前述の11個のカテゴリごとの出現個数
  - Linuxコマンドによって度数分布一覧を作成した方法について説明すること.
    ※作成方法の説明は省略せずに，すべて記述すること.
- report-3-学籍番号-log.txt
  - 前述の100個の乱数を保存したテキストファイル

提出に関して

- 全提出ファイルを1つのフォルダに格納し，
  そのフォルダをzip圧縮したファイルを提出せよ．
- 指定フォルダ名：**report-3-学籍番号**
  - 「report-3-0123456789」の形式で，学籍番号を各自のものに置き換えること．
- 指定zipファイル名：**report-3-学籍番号.zip**
  ⇧このファイルを提出する．
- zipの中のフォルダ内のファイル構成：
  - **report-3-学籍番号.txt**
  - **report-3-学籍番号-log.txt**
- 提出期限は，授業支援システム参照
- 提出方法：授業支援システムから提出
- 提出ファイルの中に，学籍番号と氏名を明記のこと
  - プログラムではコメントの中に書くこと
- 注意点
  - 各種指示を守っていない場合(学籍番号と氏名が書かれていない，ファイル名が間違っている，
    指定されていないファイル形式で提出している等)は減点する．
  - 今回指定した環境でコンパイルできないプログラムは採点しない(0点)



zipファイル，内部フォルダ，
提出ファイル2種の構成