

計算物理学II (第3回)

前回授業に関するコメント

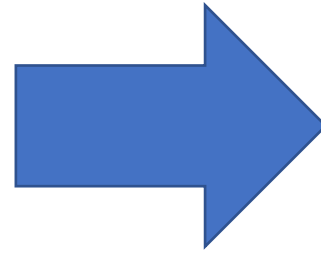
- Ubuntuで日本語入力ができない
<https://www.u.tsukuba.ac.jp/faq/#linux>
- manabaの問題4 について
- 講義資料
スマホで見ている人もいましたが大きなモニタもあるので
Linuxでpdfを表示してみても

今回の内容

- テキストエディタ(emacs)の使い方
- テキスト処理に関連するコマンドの演習(grep,sed,awkなど)
- リダイレクト・パイプライン
- ファイル、ディレクトリの圧縮・解凍

テキスト処理の例

```
hinojara.nobuo.ga@icho:~/compphys2/3rd$ cat x2.txt
calculate y = x**2
x = 0.0 y = 0.0
calculate y = x**2
x = 1.0 y = 1.0
calculate y = x**2
x = 2.0 y = 4.0
calculate y = x**2
x = 3.0 y = 9.0
calculate y = x**2
x = 4.0 y = 16.0
calculate y = x**2
x = 5.0 y = 25.0
calculate y = x**2
x = 6.0 y = 36.0
calculate y = x**2
x = 7.0 y = 49.0
calculate y = x**2
x = 8.0 y = 64.0
calculate y = x**2
x = 9.0 y = 81.0
calculate y = x**2
x = 10.0 y = 100.0
```



```
hinojara.nobuo.ga@icho:~/compphys2/3rd$ cat x2data.txt
0.0 0.0
1.0 1.0
2.0 4.0
3.0 9.0
4.0 16.0
5.0 25.0
6.0 36.0
7.0 49.0
8.0 64.0
9.0 81.0
10.0 100.0
```

今回の目標はこのようなテキストの作成・

テキストファイルから必要な情報だけを抜き出すことができるようになること

emacsの使い方

emacsの起動

テキスト編集はemacs(イーマックス)というエディタで行う。

TeXソースファイルやプログラムを書くのに使いますので基本的な使い方を覚えてください。

- ・ 起動は**emacs &**というコマンドで起動。別ウィンドウで起動する。
- ・ &をつけない場合はemacsが起動中(実行中)は端末が使えない。

(コマンドのあとに&をつけるとバックグラウンドでコマンドが実行できる)

emacsの後に開きたいファイル名を指定できる。ファイルが存在しない場合は新規作成となる

例えば test.txtというファイルを作成して編集したい場合は

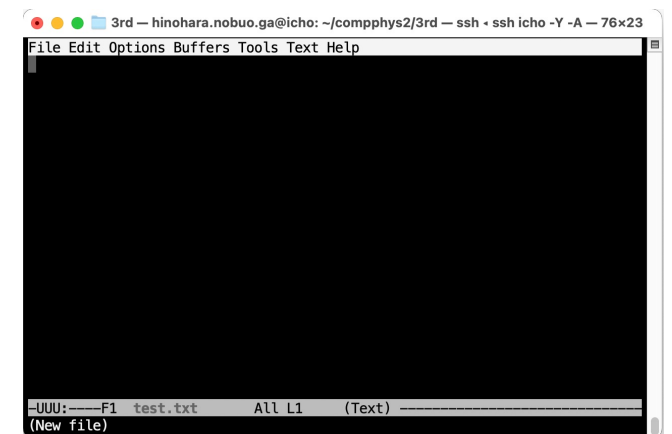
emacs test.txt &

-nwオプション(no windows)をつけると端末内でemacsが起動

(ssh接続の場合はネットワーク速度が遅いと別ウィンドウは重くて使い物にならないため-nwをつけるほうがよい)

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ emacs -nw test.txt
```

ウィンドウ全体がemacsの入力画面となる
(起動中はシェルは表示されなくなり使えない)



emacsの使い方・ファイルの保存

キーボードを使ってテキストを入力できる
後の演習で使うので左のような数字と英単語(何でもよい)
を10行ほど適当に入力。

ファイルの保存は C-x C-s

(Ctrlキーを押しながらxキーを押し、xキーを離し、
Ctrlを押しながらsを押して、sキーを話す
2つの動作の間Ctrlは押しっぱなしでOK)
入力中は頻繁にC-x C-sを押して随時保存をする。

保存されると下部のメッセージ欄に
"Wrote ファイルの絶対パス" が表示

```
File Edit Options Buffers Tools Text Help
1 physics
2 mathematics
3 mechanics
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
10 rotation
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum mechanics
16 electromagnetics
17 statistics
18 atom
19 molecule
20 science

-UUU:**--F1 test.txt All L21 (Text) -----
```

開いているファイル名

```
--UU--:----F1 test.txt Bot L21 (Text) -----
Wrote /home/hinohara.nobuo.ga/compphys2/3rd/test.txt
```

emacsの終了は C-x C-c 。シェルに制御が戻る。
保存せず終了しようとするすると保存するかを下部で聞かれるので
y (yes)かn (no)を入力

```
--UU--:**--F1 test.txt Bot L23 (Text) -----
Save file /home/hinohara.nobuo.ga/compphys2/3rd/test.txt? (y, n, !, ., q, C-r, d or C-h)
```

最低限必要なemacsの使い方

- 起動

- シェルで **emacs ファイル名 &** (または **emacs ファイル名 -nw**)
- ファイル名を指定せず **emacs & (-nw)** だけで起動し、**C-x C-f**で開くファイルを選んでもよい(新規ファイル・既存ファイル)

- 編集

- キーボードで英数字入力
- 日本語入力。今使っているコンピュータの日本語入力が使えないはず。
- Linux側の日本語入力を使う場合は **C-¥**で切り替え

- 保存

- **C-x C-s** (Ctrlキーを押しながらx、sを順番に押す)

- 終了

- **C-x C-c** (保存しないで終了しようとするすると保存するか聞かれる)

emacsの使い方

- **C-x C-f**: 新しいファイルを開く (下のメッセージ欄でパスを指定)
- **C-s** ファイル内を検索(下向き), **C-r** ファイル内を検索(上向き)
(メッセージにI-search: と出るので続けて検索したい文字列を入力)
次の検索候補に移動するときはC-sまたはC-rを何度も入力
- **C-g** コマンドのキャンセル(emacs使用中に入力できなかつたりしたときに使うとよい)

以下はシェルのコマンド入力の時と同じ

- **C-a** でカーソルを行の先頭に移動 **C-e** でカーソルを行の最後に移動
- **C-k** でカーソルのある行のカーソルより右側を削除(カット)
- **C-u** でカーソルのある行のカーソルより左側を削除(カット)
- **C-y** 貼り付け。C-kまたはC-uで削除されたものを貼り付ける(ヤंक)
- カーソルの移動: **C-p** (上) **C-n** (下) **C-b** (左) **C-f** (右)

emacsの使い方

- **M-%**: (Metaキー(Esc)を押して離した後% (shift + 5)を押す) 文字列の置換。メッセージ欄にQuery replace: と出るので置換する文字列を入力してEnter
- 置換後の文字列を入力してEnter
- ファイルを上から検索するので置換したい場合はy, しない場合はnをそれぞれ入力
- **C-l**: (エル)カーソルのある行が中央に来るように表示領域を移動

emacsのバックアップファイル

- emacsでファイル(test.txt)を編集して保存すると保存前のファイルが test.txt~ という名前でディレクトリに残る。元に戻したいときはmvコマンドを使って戻すことができる。
- 一個前のバージョンしか保存されない。

```
File Edit Options Buffers Tools Text Help
0 test
1 physics 一行追加して保存してみる
2 mathematics
3 mechanics
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
10 rotation
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum mechanics
16 electromagnetics
17 statistics
18 atom
19 molecule
-UU-:----F1 test.txt Top L1 (Text) -----
Wrote /home/hinohara.nobuo.ga/compphys2/3rd/test.txt
```

test.txt~に保存前のデータが保存される *は任意の文字列

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ emacs -nw test.txt
hinohara.nobuo.ga@icho:~/compphys2/3rd$ ls test.txt*
test.txt test.txt~
hinohara.nobuo.ga@icho:~/compphys2/3rd$ diff test.txt test.txt~
1d0
< 0 test
hinohara.nobuo.ga@icho:~/compphys2/3rd$
```

1行目だけが異なる (diffコマンドは2つのファイルの差分を表示)
diff ファイル1 ファイル2

その他のテキストエディタ

- emacsよりも高速に動作するテキストエディタとしてvi (ヴィーアイ)、またはvim(ヴィム)がある。使い方が少し特徴的。ファイルの一部だけの修正などで便利。
- 最近ではVisual Studio Codeという多機能なコードエディタがプログラムを書くのに主流になりつつある。このエディタではsshの設定を行うことによりリモートのファイルの編集も可能。
- sshでリモートのコンピュータにあるファイルの(小規模な)編集ではemacsまたはviが主流。

テキスト処理に関するコマンド

- head, tail: ファイルの先頭、末尾の指定行だけ表示
- wc: ファイルの文字数、行数などの情報を表示
- sort: 指定列で並び替え
- grep: ファイルの内容検索
- sed: 行ごとにファイルの内容変換など
- awk: 行ごとにファイルの内容の一部表示など
- diff: 2つのファイルの差分を表示する
- uniq: 重複行を削除
- cut: 一部の列だけを取り出す

ファイルの一部だけを表示

emacsで作ったファイルを使っていくつかのコマンドの演習を行う。

head -n 行数 ファイル名 : ファイルの上から指定行だけを表示

tail -n 行数 ファイル名 : ファイルの下から指定行だけを表示

長いファイルの一部だけを見たい場合に便利。

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ ls
test.txt
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head test.txt
0 test
1 physics
2 mathematics
3 mechanics
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 2 test.txt
0 test
1 physics
hinohara.nobuo.ga@icho:~/compphys2/3rd$
```

行数を指定しない場合は10行表示

上から2行を表示

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ tail test.txt
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum mechanics
16 electromagnetics
17 statistics
18 atom
19 molecule
20 science
hinohara.nobuo.ga@icho:~/compphys2/3rd$ tail -n 5 test.txt
16 electromagnetics
17 statistics
18 atom
19 molecule
20 science
```

-nオプションなしで
下から10行を表示

下から5行だけ表示

ファイルの行数・単語数を表示(wc)

wc ファイル名

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ wc test.txt
21  43 254 test.txt
hinohara.nobuo.ga@icho:~/compphys2/3rd$ █
```

test.txtファイルは21行、43単語、254バイトであることが表示される (word count)

ちなみにこのディレクトリにtest.txtファイルしかない場合はタブ補完が使えます。
wcとスペースを入力後タブキーで自動的にtest.txtが補完されます。(他のコマンドの場合も)

並び替え (sort)

sort ファイル名

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sort test.txt
0 test
1 physics
10 rotation
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum mechanics
16 electromagnetics
17 statistics
18 atom
19 molecule
2 mathematics
20 science
3 mechanics
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
```

先頭の文字順に並び替え

sort -n ファイル名

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sort -n test.txt
0 test
1 physics
2 mathematics
3 mechanics
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
10 rotation
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum mechanics
16 electromagnetics
17 statistics
18 atom
19 molecule
20 science
```

-nオプションで数値順で並び替え
(もともとと同じ並び)

並び替え (sort)

sort -r ファイル名

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sort -r test.txt
9 method
8 derivative
7 integration
6 analysis
5 computer
4 equation
3 mechanics
20 science
2 mathematics
19 molecule
18 atom
17 statistics
16 electromagnetics
15 quantum mechanics
14 factor
13 Maxwell
12 ratio
11 divergence
10 rotation
1 physics
0 test
```

文字列として逆順にソートして出力

sort -nr ファイル名 (-n -rや -rnもOK)

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sort -nr test.txt
20 science
19 molecule
18 atom
17 statistics
16 electromagnetics
15 quantum mechanics
14 factor
13 Maxwell
12 ratio
11 divergence
10 rotation
9 method
8 derivative
7 integration
6 analysis
5 computer
4 equation
3 mechanics
2 mathematics
1 physics
0 test
```

数値で逆順にソートして出力

並び替え (sort)

sort -k 列数 ファイル名

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sort -k 2 test.txt
13 Maxwell
6 analysis
18 atom
5 computer
8 derivative
11 divergence
16 electromagnetics
4 equation
14 factor
7 integration
2 mathematics
3 mechanics
9 method
19 molecule
1 physics
15 quantum mechanics
12 ratio
10 rotation
20 science
17 statistics
0 test
```

2列目の値で並び替えて出力。大文字のMが先頭に来て小文字のaから順に並んでいる。

文字列検索(grep)

grep 文字列 ファイル名 でファイルの中から文字列を含む行のみを出力
(global regular expression print、グレップ)

test.txtで”cs”を含む行のみを出力

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ grep cs test.txt  
1 physics  
2 mathematics  
3 mechanics  
15 quantum mechanics  
16 electromagnetics  
17 statistics
```

オプション

-n 行番号をつける

-v 文字列を含まない行だけを出力

-i 文字列の大文字と小文字を区別しない

grepでは**正規表現**(regular expression)を使うことができる。正規表現の詳細は省略するが色々な文字列を一つの文字列で表現できる。grep 正規表現などで検索すると多くの情報を見つけることができる。

文字列の置換(sed)

sedとは：Stream Editor。読み込んだファイルに対して処理を行い、結果を標準出力に出力する。(結果をファイルに保存したい場合はリダイレクトを使う)

文字列の置換 `sed 's/置換前文字列/置換後文字列/g' ファイル名`

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed 's/mechanics/theory/' test.txt
0 test
1 physics
2 mathematics
3 theory
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
10 rotation
11 divergence
12 ratio
13 Maxwell
14 factor
15 quantum theory
16 electromagnetics
17 statistics
18 atom
19 molecule
20 science
```

mechanicsだったところがtheoryに置換されている

gフラグがない場合は一行に複数のmechanicsが含まれている場合は最初の一つだけ置換してその行の処理は終了する。

一行にある複数のmechanicsを全て置換する場合はgフラグをつける。

行の削除(sed)

特定の行を削除：**sed '行番号d' ファイル名**

行1から行2までを削除：**sed '行番号1、行番号2d' ファイル名**

文字列を含む行を削除：**sed '/文字列/d' ファイル名**

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '2,4d' test.txt
0 test
4 equation
5 computer
6 analysis
7 integration
8 derivative
9 method
```

2-4行目を削除

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '3d' test.txt
0 test
1 physics
3 mechanics
4 equation
5 computer
6 analysis
```

3行目を削除

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '/ics/d' test.txt
0 test
4 equation
5 computer
6 analysis
7 integration
8 derivative
```

文字列icsを含む行を削除

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '2,$d' test.txt
0 test
```

最終行は\$でも表すことができる
2行目から最終行までを削除

行の挿入(sed)

ある行の直前に文字列を挿入：**sed '行番号i 文字列' ファイル名**
ある行の直後に文字列を挿入：**sed '行番号a 文字列' ファイル名**

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '2i biology' test.txt
0 test
biology
1 physics
2 mathematics
3 mechanics
4 equation
```

2行目(1 physics)の直前にbiologyの文字列からなる行を挿入

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ sed '2a biology' test.txt
0 test
1 physics
biology
2 mathematics
3 mechanics
4 equation
```

2行目(1 physics)の直後にbiologyの文字列からなる行を挿入

行の上書き (sed)

ある行を文字列で上書き : **sed** '行番号c 文字列' ファイル名

```
[hinozara.nobuo.ga@icho:~/compphys2/3rd$ sed '2c 1 biology' test.txt
0 test
1 biology
2 mathematics
3 mechanics
4 equation
5 computer
6 analysis
7 integration
```

2行目(1 physics)を'1 biology'文字列で上書き

列の取り出し(awk)

- awk(オーク)とは：テキスト処理を行うプログラミング言語。
- ファイルを読み込み1行ずつ処理が行われる
- 書式は **awk パターン '{アクション}' ファイル名**
- よく使うのは列の取り出し。複雑な処理も可能だがPythonなど他の言語を使うことが多い。

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ awk '{print $2}' test.txt
test
physics
mathematics
mechanics
equation
computer
analysis
```

2列目を出力

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ awk '{print $2, $1}' test.txt
test 0
physics 1
mathematics 2
mechanics 3
equation 4
computer 5
analysis 6
```

2列目、1列目の順に出力
レコードは","で区切る

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ awk '{print $1, $2, $1}' test.txt
0 test 0
1 physics 1
2 mathematics 2
3 mechanics 3
4 equation 4
5 computer 5
6 analysis 6
```

1列目、2列目、1列目の順に出力

その他のテキスト処理のコマンド

- diff: 2つのファイルの差分を表示する
- uniq: 重複行を削除
- cut: 一部の列だけを取り出す

リダイレクト・パイプライン

リダイレクト(標準出力)

- コマンド実行の基本的な動作：
 - コマンドを入力→端末の画面の次の行に結果が出力
 - コマンドの出力は**標準出力(stdout; standard output)**に出力されるようになっている。
 - シェルで**標準出力=画面への出力**という設定になっている
- **標準出力のリダイレクト**
 - 標準出力を画面ではなく指定したファイルに出力
 - 書き方 **コマンド(オプション、ファイルなど含む) > 出力ファイル**

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 2 test.txt
0 test
1 physics
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 2 test.txt > head.txt
hinohara.nobuo.ga@icho:~/compphys2/3rd$ cat head.txt
0 test
1 physics
```

headコマンドを実行すると画面(標準出力)に結果が出力される

出力結果を head.txtにリダイレクト
画面上には何も表示されない

head.txtファイルの中身をcatで確認すると head -n 2 test.txtの結果が保存されている

リダイレクト(標準出力)

標準出力のリダイレクトをすでに存在するファイルに対して行うと上書きされる。
上書きではなく追記をしたい場合は >> を使う

```
hinothara.nobuo.ga@icho:~/compphys2/3rd$ head -n 2 test.txt > head.txt
hinothara.nobuo.ga@icho:~/compphys2/3rd$ cat head.txt
0 test
1 physics
hinothara.nobuo.ga@icho:~/compphys2/3rd$ tail -n 2 test.txt > head.txt
hinothara.nobuo.ga@icho:~/compphys2/3rd$ cat head.txt
19 molecule
20 science
hinothara.nobuo.ga@icho:~/compphys2/3rd$ head -n 2 test.txt >> head.txt
hinothara.nobuo.ga@icho:~/compphys2/3rd$ cat head.txt
19 molecule
20 science
0 test
1 physics
```

head.txtファイルを作る

すでに存在するファイルに対して
リダイレクトすると上書きされる

追記するには>の代わりに>>を使う

tailの実行結果の下にheadの実行結果が追記される

リダイレクト(標準エラー出力)

- コマンドを実行すると画面にエラーメッセージが出力されることがある。
- コマンドのエラーメッセージは標準出力ではなく**標準エラー出力(stderr; standard error)**に出力されるようになっている。
- シェルでは**標準エラー出力 = 画面への出力**に設定されている
- 標準エラー出力をファイルへリダイレクトするには **2>** を使う

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ a
a: コマンドが見つかりません
hinohara.nobuo.ga@icho:~/compphys2/3rd$ a > error
a: コマンドが見つかりません
hinohara.nobuo.ga@icho:~/compphys2/3rd$ cat error
hinohara.nobuo.ga@icho:~/compphys2/3rd$ a 2> error
hinohara.nobuo.ga@icho:~/compphys2/3rd$ cat error
a: コマンドが見つかりません
```

存在しないコマンドを実行すると
エラーメッセージが出力される

標準出力をリダイレクトしても
エラーメッセージは画面に出力され
errorファイルには何も書かれていない

標準エラー出力をリダイレクトすると
画面には何も出力されなくなり、
errorファイルにエラーメッセージが
書き込まれる

リダイレクト(標準入力)

- コマンドの中には**標準入力(stdin; standard input)**から値を受け取って結果を表示するものがある。
 - シェルで**標準入力=キーボード**からの入力となっている
 - 標準入力をファイルにリダイレクトするには **コマンド < ファイル名** とする
- catコマンド**：標準入力(キーボードからの入力)からの入力を標準出力(画面)に表示する
終了するにはC-d (または強制終了のC-c)

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ cat ファイルを指定せずに実行
123 キーボードで123と打ってEnter(標準入力)
123 画面上に123が表示される(標準出力)
456 キーボードで456と打ってEnter(標準入力)
456 画面上に456が表示される(標準出力)
C-dで終了 hinohara.nobuo.ga@icho:~/compphys2/3rd$ cat < head.txt
19 molecule 標準入力をファイルにリダイレクト
20 science
0 test
1 physics
```

ファイルの中身が標準出力に表示される

例えば **cat < head.txt > head.out** とすると head.txtの内容をhead.outに出力となる。

リダイレクトの応用

標準入力をinputファイルから読み込み
標準出力をoutputファイルに書き出し、
標準エラー出力をerrorファイルに書き出したい場合

コマンド < input > output 2> error

標準出力と標準エラー出力両方をoutputファイルに書き出す場合

コマンド > output 2>1&

標準出力をどこにも出力したくない場合(出力が大量でいらない場合など)

コマンド > /dev/null

/dev/nullファイルに書き込まれたデータはすべて破棄される。(nullデバイス)

パイプライン

- コマンドの出力結果(標準出力)を他のコマンドの標準入力に渡す
- コマンド1 | コマンド2 とするとコマンド1の標準出力がコマンド2の標準入力に渡される (縦棒は右上のBackspaceの隣のキーをshiftキーと一緒に押す)

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 5 test.txt
```

```
0 test
```

```
1 physics
```

```
2 mathematics
```

```
3 mechanics
```

```
4 equation
```

test.txtファイルの先頭から5行を標準出力(画面)に表示

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 5 test.txt | sort -k 2
```

```
4 equation
```

```
2 mathematics
```

```
3 mechanics
```

```
1 physics
```

```
0 test
```

アルファベット順に
並び替わる

test.txtの先頭から5行をsortコマンドの標準入力にわたす
sortコマンドでは2列目で並び替えを行った結果を
標準出力(画面)に表示

```
hinohara.nobuo.ga@icho:~/compphys2/3rd$ head -n 5 test.txt | sort -k 2 | head -n 1
```

```
4 equation
```

sortコマンドの出力をheadコマンドに渡し、先頭の1行だけを標準出力に
表示

パイプラインの応用

- 出力が長いコマンドの表示

historyコマンド。何百行も表示される。**history | less** とすることで先頭から順番に表示することができる

```
523 2021-10-15T15:40:25+0900 head -n 2 test.txt
524 2021-10-15T15:40:38+0900 head -n 2 test.txt > head.txt
525 2021-10-15T15:40:40+0900 cat head.txt
526 2021-10-15T15:44:20+0900 head -n 2 test.txt > head.txt
527 2021-10-15T15:44:23+0900 cat head.txt
528 2021-10-15T15:44:29+0900 tail -n 2 test.txt > head.txt
529 2021-10-15T15:44:31+0900 cat head.txt
530 2021-10-15T15:44:40+0900 head -n 2 test.txt >> head.txt
531 2021-10-15T15:44:41+0900 cat head.txt
532 2021-10-15T15:46:04+0900 a
533 2021-10-15T15:46:07+0900 a > error
534 2021-10-15T15:46:11+0900 cat error
535 2021-10-15T15:46:14+0900 a 2> error
536 2021-10-15T15:46:16+0900 cat error
537 2021-10-15T15:48:47+0900 cat
538 2021-10-15T15:49:06+0900 cat < head.txt
539 2021-10-15T15:51:42+0900 head -n 5 test.txt
540 2021-10-15T15:51:45+0900 head -n 5 test.txt | sort -k 2
541 2021-10-15T15:51:49+0900 head -n 5 test.txt | sort -k 2 | head -n 1
542 2021-10-15T15:55:06+0900 history
hinohara.nobuo.ga@icho:~/compphys2/3rd$
```

複数のコマンドを順に適用して処理ができるのでプログラムを書かなくても簡単な処理が可能

URLからファイルをダウンロードする(wget)

wget インターネットアドレス(URL) : 指定したURLのファイルをダウンロードする

授業のホームページからx2.txtをダウンロード

```
hinothara.nobuo.ga@icho01:~/compphys2/3rd$ wget https://wwnucl.ph.tsukuba.ac.jp/~hinothara/compphys2-23/doc/x2.txt
--2023-10-19 11:03:18-- https://wwnucl.ph.tsukuba.ac.jp/~hinothara/compphys2-23/doc/x2.txt
wwnucl.ph.tsukuba.ac.jp (wwnucl.ph.tsukuba.ac.jp) をDNSに問いあわせています... 130.158.70.169
wwnucl.ph.tsukuba.ac.jp (wwnucl.ph.tsukuba.ac.jp)|130.158.70.169|:443 に接続しています... 接続しました。
HTTPによる接続要求を送信しました、応答を待っています... 200 OK
長さ: 462 [text/plain]
`x2.txt' に保存中

x2.txt          100%[=====>]          462  --.-KB/s   in 0s

2023-10-19 11:03:18 (128 MB/s) - `x2.txt' ^保存完了 [462/462]

hinothara.nobuo.ga@icho01:~/compphys2/3rd$ ls          保存完了
x2.txt
```

カレントディレクトリに保存される

ファイル・ディレクトリの圧縮・解凍

ディレクトリの圧縮

Windowsでは**zip**がよく使われるがLinuxでは**.tgz(または.tar.gz)**がディレクトリの圧縮形式としてよく使われる。

.tarはディレクトリを一つのファイルにアーカイブした拡張子、.gzはgzipで圧縮した拡張子。これらを合わせて .tar.gzまたは.tgzという拡張子が用いられる。

tarコマンドによる圧縮：**tar czvf 圧縮ファイル 圧縮するディレクトリ**

オプションczvfの順番は何でも良い c: 圧縮ファイルを作成 z: gzip圧縮 v: 結果を画面に表示 f: ファイルに圧縮 vはなくてもよい。zがないと圧縮されない(tarアーカイブ)。圧縮しない場合は.tarという拡張子を使う

```
hinohara.nobuo.ga@icho:~/compphys2$ ls
3rd/
hinohara.nobuo.ga@icho:~/compphys2$ ls 3rd/
error head.txt test.txt x2.txt x2data.txt
hinohara.nobuo.ga@icho:~/compphys2$ tar czvf 3rd.tgz 3rd
3rd/
3rd/x2.txt
3rd/error
3rd/head.txt
3rd/test.txt
3rd/x2data.txt
hinohara.nobuo.ga@icho:~/compphys2$ ls
3rd/ 3rd.tgz
```

ディレクトリの圧縮はディレクトリの外から行う

3rdディレクトリを3rd.tgzファイルに圧縮

ディレクトリの解凍

- tarコマンドによる解凍：**tar zxvf 圧縮ファイル**
- x:解凍のオプション

```
hinohara.nobuo.ga@icho:~/compphys2$ mv 3rd 3rd.bak
hinohara.nobuo.ga@icho:~/compphys2$ ls
3rd.bak/ 3rd.tgz
hinohara.nobuo.ga@icho:~/compphys2$ tar zxvf 3rd.tgz
3rd/
3rd/x2.txt
3rd/error
3rd/head.txt
3rd/test.txt
3rd/x2data.txt
hinohara.nobuo.ga@icho:~/compphys2$ ls
3rd/ 3rd.bak/ 3rd.tgz
```

すでに3rdというディレクトリがあるので
名前を変えておく

3rd.tgzをこのディレクトリに解凍

解凍したディレクトリが現れる

この他にbz2圧縮形式による .tar.bz2 (tbz) という拡張子もLinuxでは用いられる。
zの代わりにjオプションを用いる

第3回課題

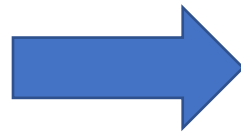
- ・ manabaから提出(提出任意・成績には反映しません)

1. historyコマンドであなたが端末にこれまで(の一定期間)に入力したコマンドの一覧が出力されます。これまでにlsコマンドを入力した回数のみを標準出力に出すにはどのようなコマンドを入力すればよいですか？
(grep, パイプライン, wc, awkなどを使うと1行で書けます)。
これまでに何回lsを入力していましたか？
(タイプミスも含めてlsを含むコマンドを打った場合、でOK)

第3回課題

2. 次のファイルx2.txtを<https://wwwnucl.ph.tsukuba.ac.jp/~hinohara/compphys2-23/doc/x2.txt>からダウンロードして表示してみましょう。(wgetコマンドでダウンロード)
y=x²を計算したのですが、数字以外の情報が多くて扱いにくいです。
x2.txtから右のような数字のみ取り出したx2data.txtファイルを作成するにはリダイレクト、パイプライン、grep、sed、awkなどを使って(一行のコマンドで)どのように書けますか。

```
hinohara.nobuo.ga@physicslinux1:~/3rd$ cat x2.txt
calculate y = x**2
x = 0.0 y = 0.0
calculate y = x**2
x = 1.0 y = 1.0
calculate y = x**2
x = 2.0 y = 4.0
calculate y = x**2
x = 3.0 y = 9.0
calculate y = x**2
x = 4.0 y = 16.0
calculate y = x**2
x = 5.0 y = 25.0
calculate y = x**2
x = 6.0 y = 36.0
calculate y = x**2
x = 7.0 y = 49.0
calculate y = x**2
x = 8.0 y = 64.0
calculate y = x**2
x = 9.0 y = 81.0
calculate y = x**2
x = 10.0 y = 100.0
```



```
hinohara.nobuo.ga@physicslinux1:~/3rd$ cat x2data.txt
0.0 0.0
1.0 1.0
2.0 4.0
3.0 9.0
4.0 16.0
5.0 25.0
6.0 36.0
7.0 49.0
8.0 64.0
9.0 81.0
10.0 100.0
```