

プログラミングI 付録 プログラム実習の手順

2016年9月25日

2017年10月22日改訂版

2018年9月25日 改訂版

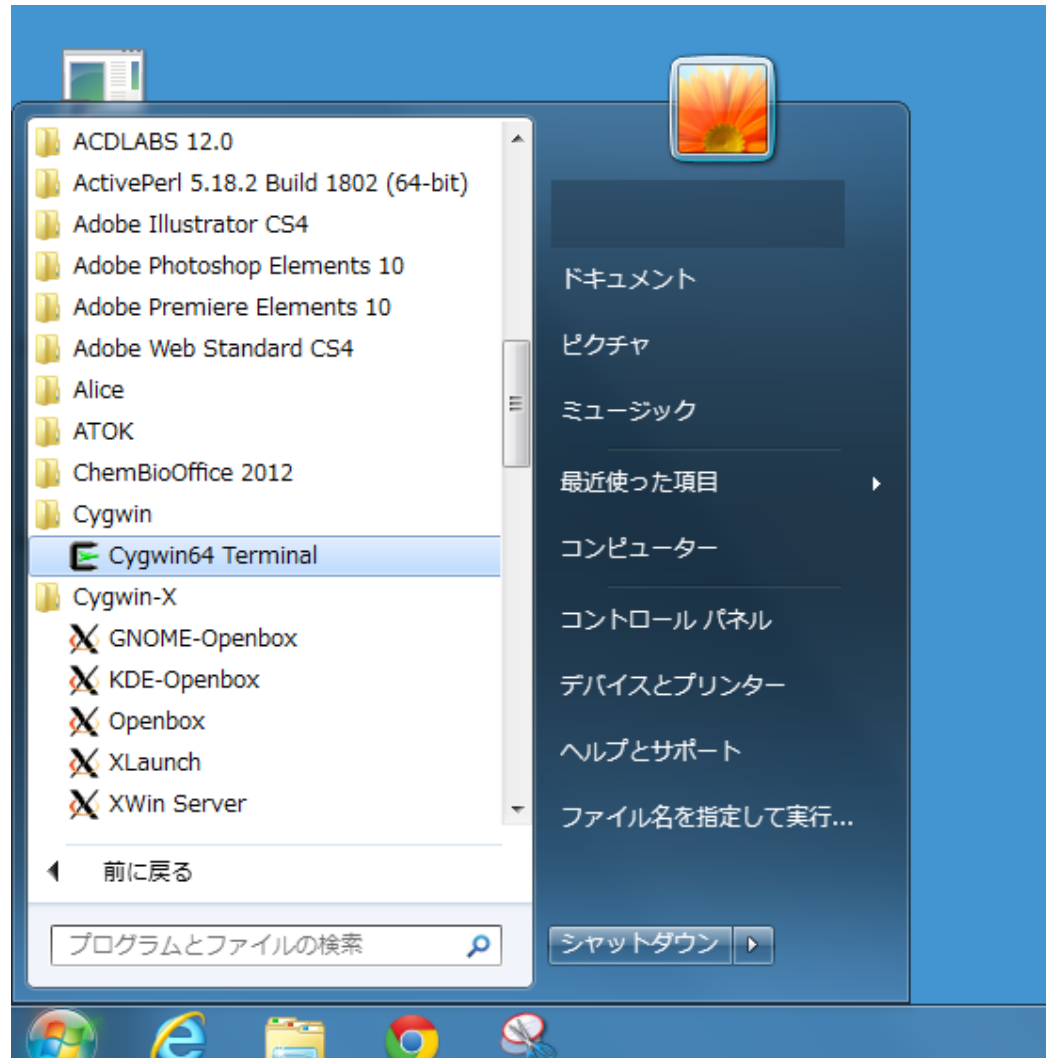
海谷 治彦

概要

- 本稿では、授業中等の実習で具体的に、どうやって操作すればよいかを解説する。
 - 慣れるまでは、このスライドを見ながら作業しましょう。
- スライド 3 から 11 までの9ステップの手順を毎回の実習毎に繰り返すこと。
 - ステップ4,5,6は繰り返す場合がある。
 - スライド中の
nantoka.c
 - は、毎回、違うファイル名に変えること。
- やってはいけないことも示す。
- ファイルの保存場所等についても言及する
- Zip の使い方も追加
- 一応、末尾にそれぞれの手順の意味について概説するが、とりあえず、作業ができてれば良い人は読み飛ばしてよい。

ステップ 1/9

- スタートメニューから, Cygwin64 Terminal を起動する.
- 以降, 開いたウィンドウはターミナルと呼ぶ.



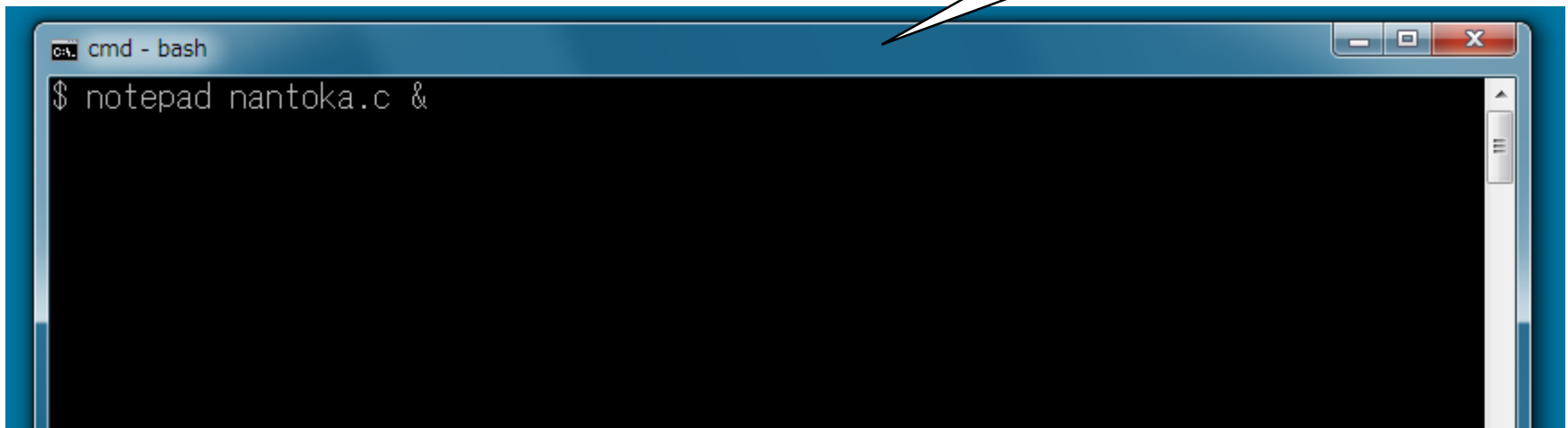
ステップ 2/9

- ターミナルのタイトルバーをクリックし、ターミナルに焦点を合わせる.

```
notepad nantoka.c &
```

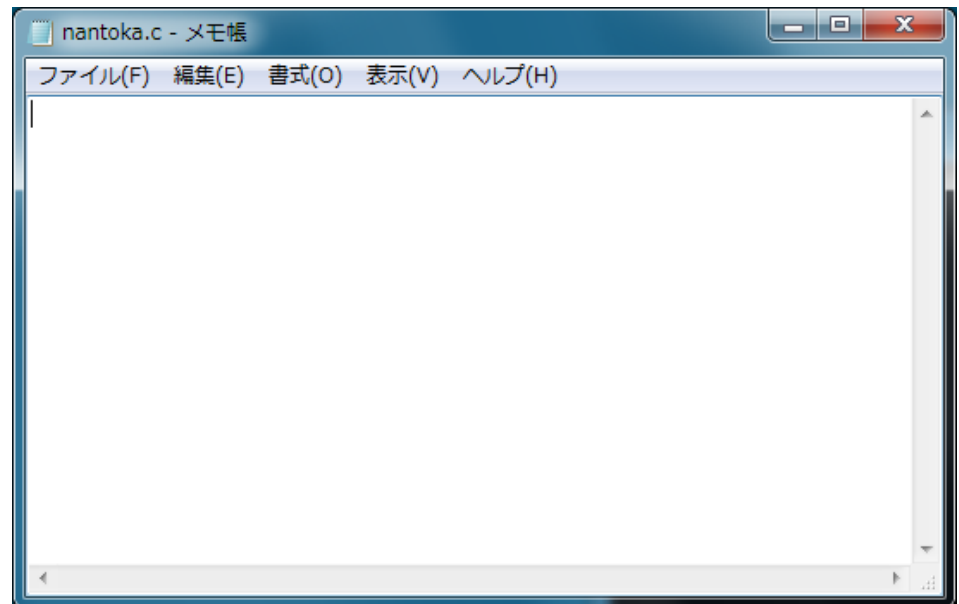
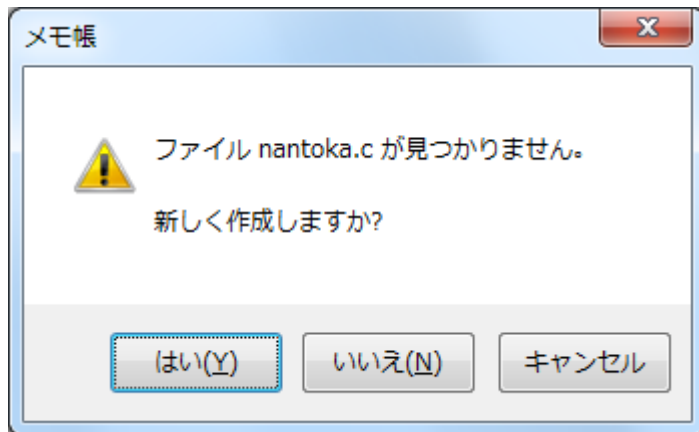
- と打ち込んで、エンターキーを押す.
 - 空白で区切るのを忘れずに.

ココがタイトルバー



ステップ 3/9

- 新しいファイルの場合，左下のような窓が出るので，「はい」を選択し，
- ソースプログラムを右下のように書き込めるようにする．
- 左下が出なければ，気にしなくてよい．



ステップ 4/9

- メモ帳のタイトルバーをクリックし、キーボードから、プログラムファイルの内容を最後まで打ち込む。
 - 下記が例.
 - ステップ6から戻ってきた際には修正を行う。
 - *どのように修正すればよいかは人間が考えるしかない！ソレがプログラミングである。*
- 通常、`#include <stdio.h>` 等から開始される。

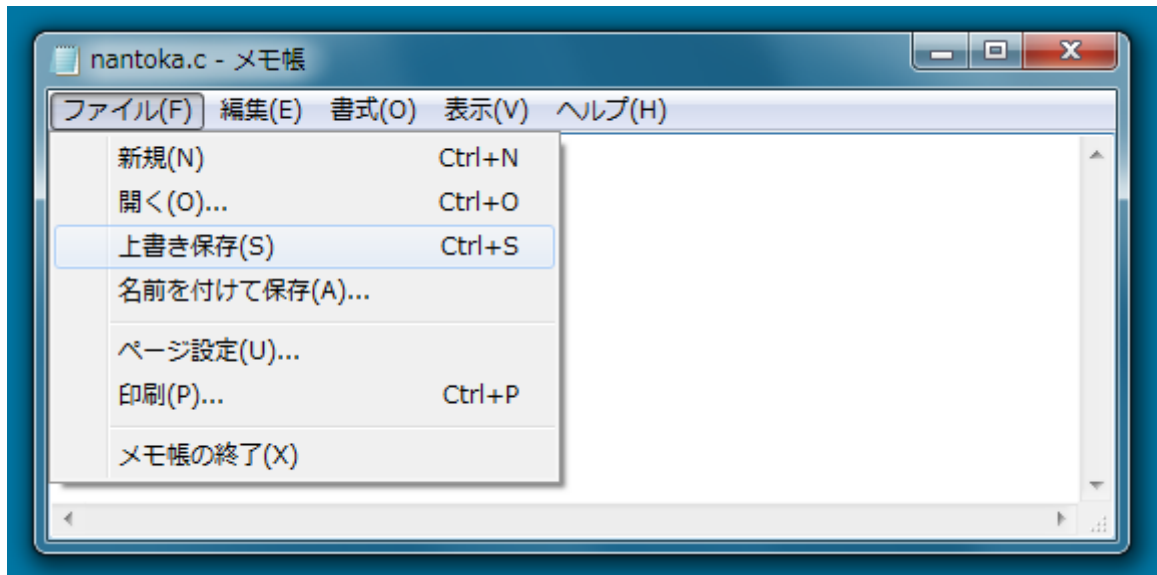
```
nantoka.c - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
#include <stdio.h>

int main(void){
    printf("hello!");
    return 0;
}
```

ココがタイトルバー

ステップ 5/9

- メモ帳で、ファイルをクリックし「上書き保存」を選び、保存する。

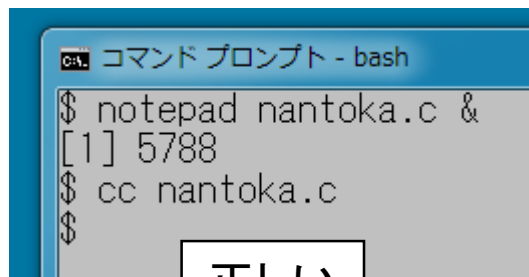


ステップ 6/9

- ターミナルのウィンドウのタイトルバーを再度クリックし,

```
cc nantoka.c
```

- とキーボードから打って, エンターを押す.
 - 正しくコンパイルできれば左のように警告は出ない.
 - ステップ7へいく.
 - プログラムに誤りがあれば右のように何か警告が出る.
 - ステップ4に戻り, プログラムを修正する.



```
コマンドプロンプト - bash
$ notepad nantoka.c &
[1] 5788
$ cc nantoka.c
$
```

正しい



```
コマンドプロンプト - bash
$ notepad nantoka.c &
[1] 7104
$ cc nantoka.c
nantoka.c: In function 'main':
nantoka.c:5:3: error: expected ';' before 'return'
    return 0;
    ^
$
```

誤り

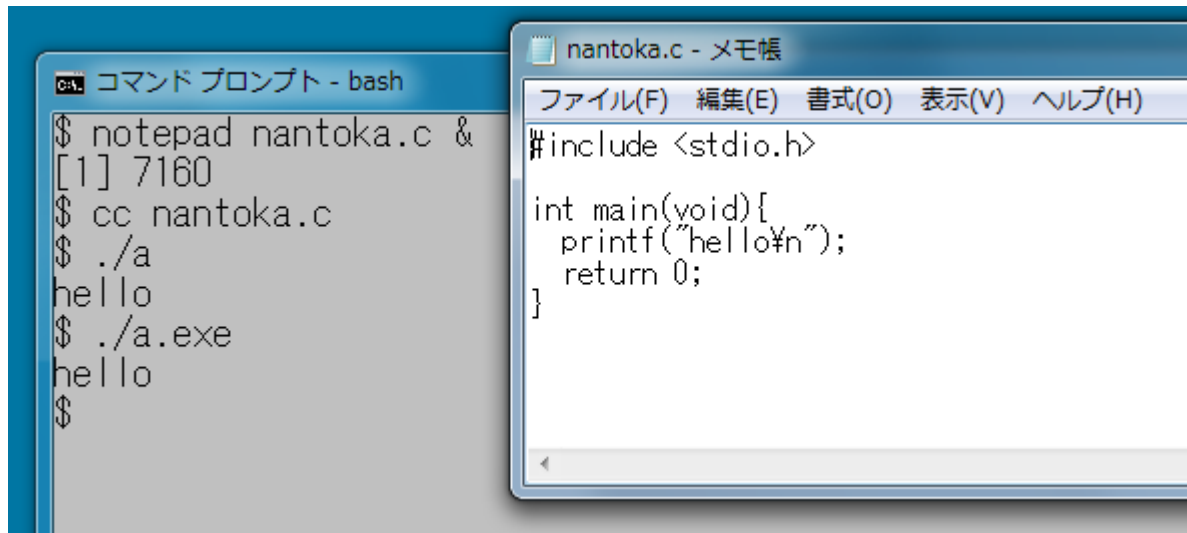
ステップ 7/9

- ターミナルのタイトルバーをクリックし、以下を打ち込み、

`./a`

エンターを押す。(`./a.exe` でもよい)

- 参考: `cc`は指定しなければ `a.exe` という名前の実行ファイルを作成する。
- 実行結果が自分の期待したものと合致していれば、ステップ8に行く。
- 結果が期待通りで無い場合、[ステップ4に戻り、プログラムを修正する。](#)



The screenshot shows two windows. The left window is a Command Prompt titled 'コマンドプロンプト - bash'. It shows the following commands and output:

```
$ notepad nantoka.c &
[1] 7160
$ cc nantoka.c
$ ./a
hello
$ ./a.exe
hello
$
```

The right window is a Notepad window titled 'nantoka.c - メモ帳'. It shows the following C code:

```
#include <stdio.h>

int main(void){
    printf("hello\n");
    return 0;
}
```

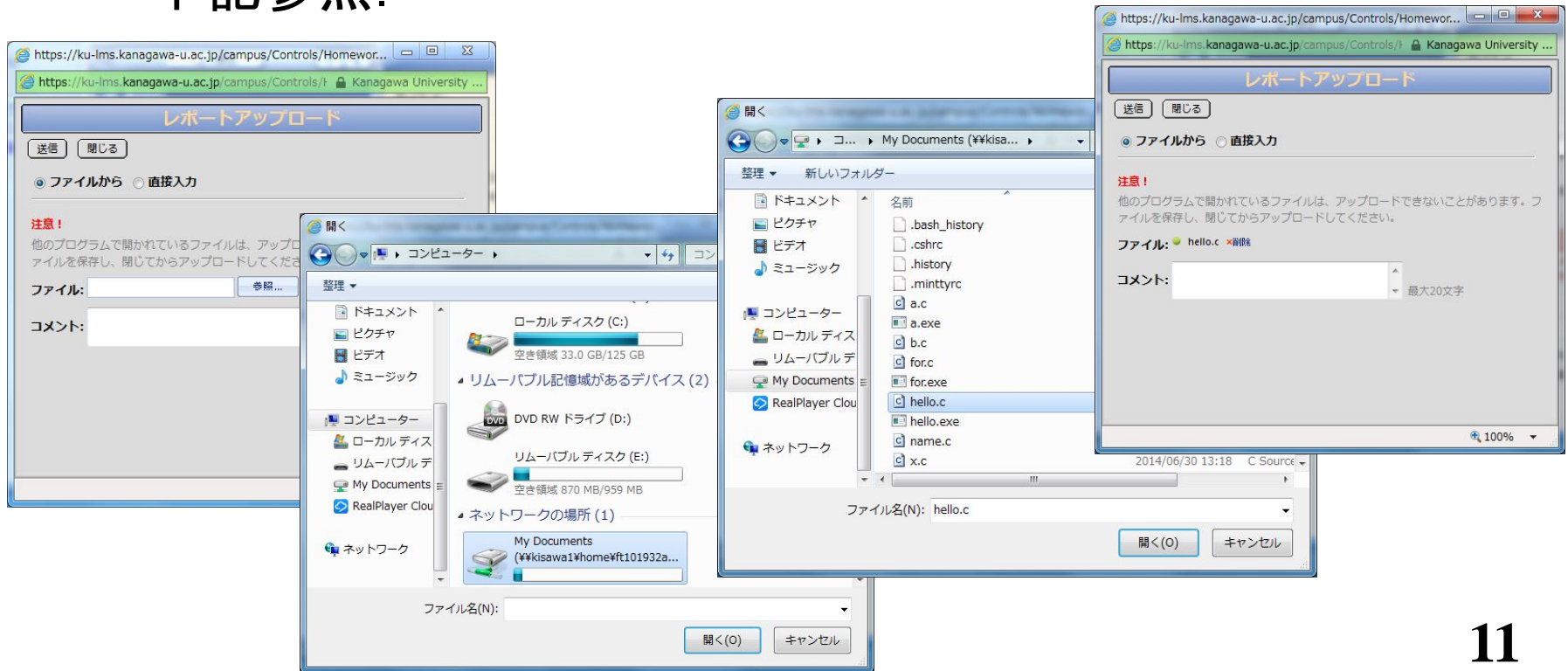
初回の実行では

- 右のようにセキュリティソフトが警告するかもしれない。
- アプリケーションの権限を変更します...を押して、実行に許可を与える。



ステップ 8/9

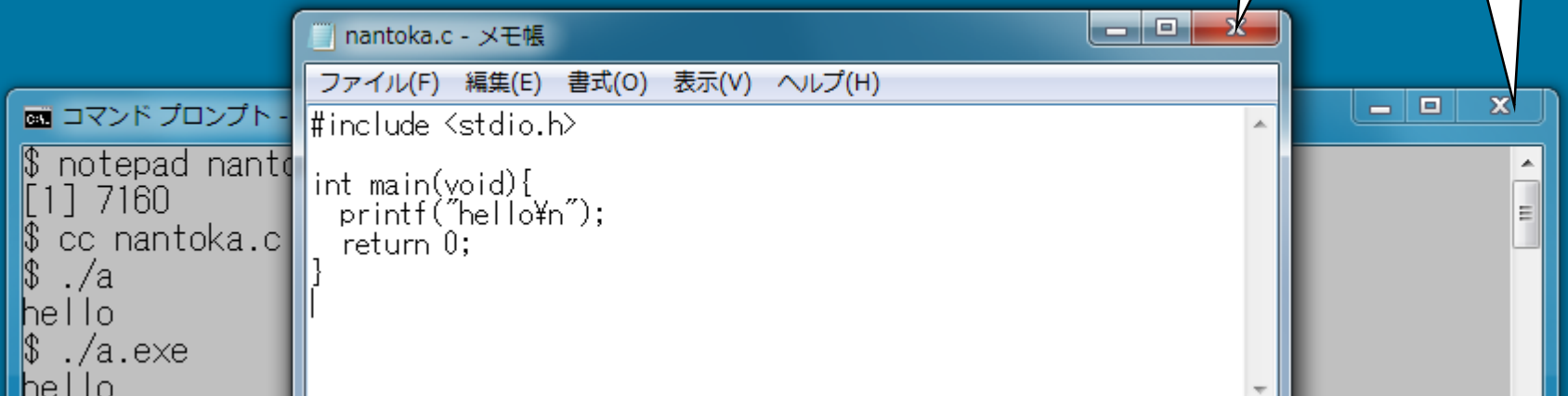
- レポート提出が無い場合は、このステップは不要.
- dotcampusの当該レポートの部分を開き、作成したソースプログラム nantoka.c を選択する.
- 通常、作成したプログラムは「ネットワークの場所」にある.
下記参照.



ステップ 9/9

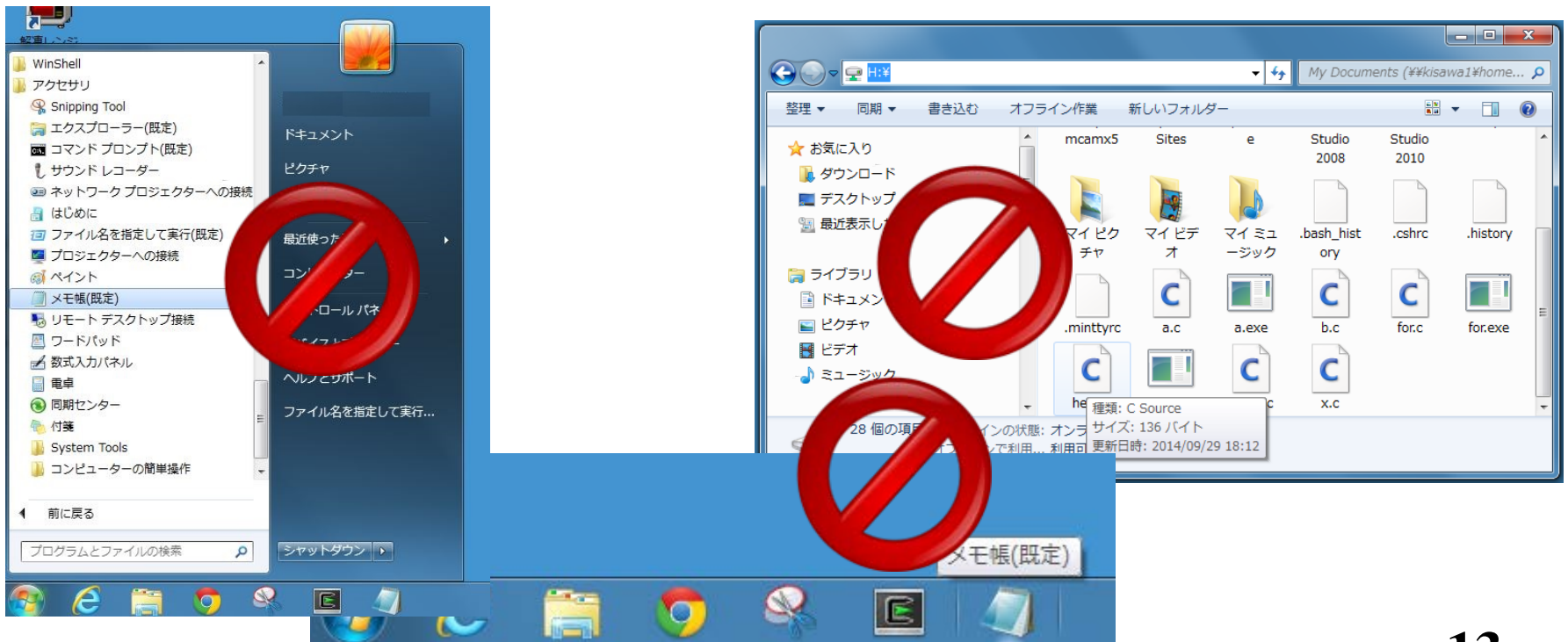
- ターミナルとメモ帳の双方のウィンドウの右端の [x] をクリックして、双方のウィンドウを閉じる.

ココを押す



やってはいけないこと

- スタートメニューからメモ帳を呼び出してはいけない。(左)
- nantoka.c 等のソースプログラムをエクスプローラー等からクリックしてはいけない。(右)

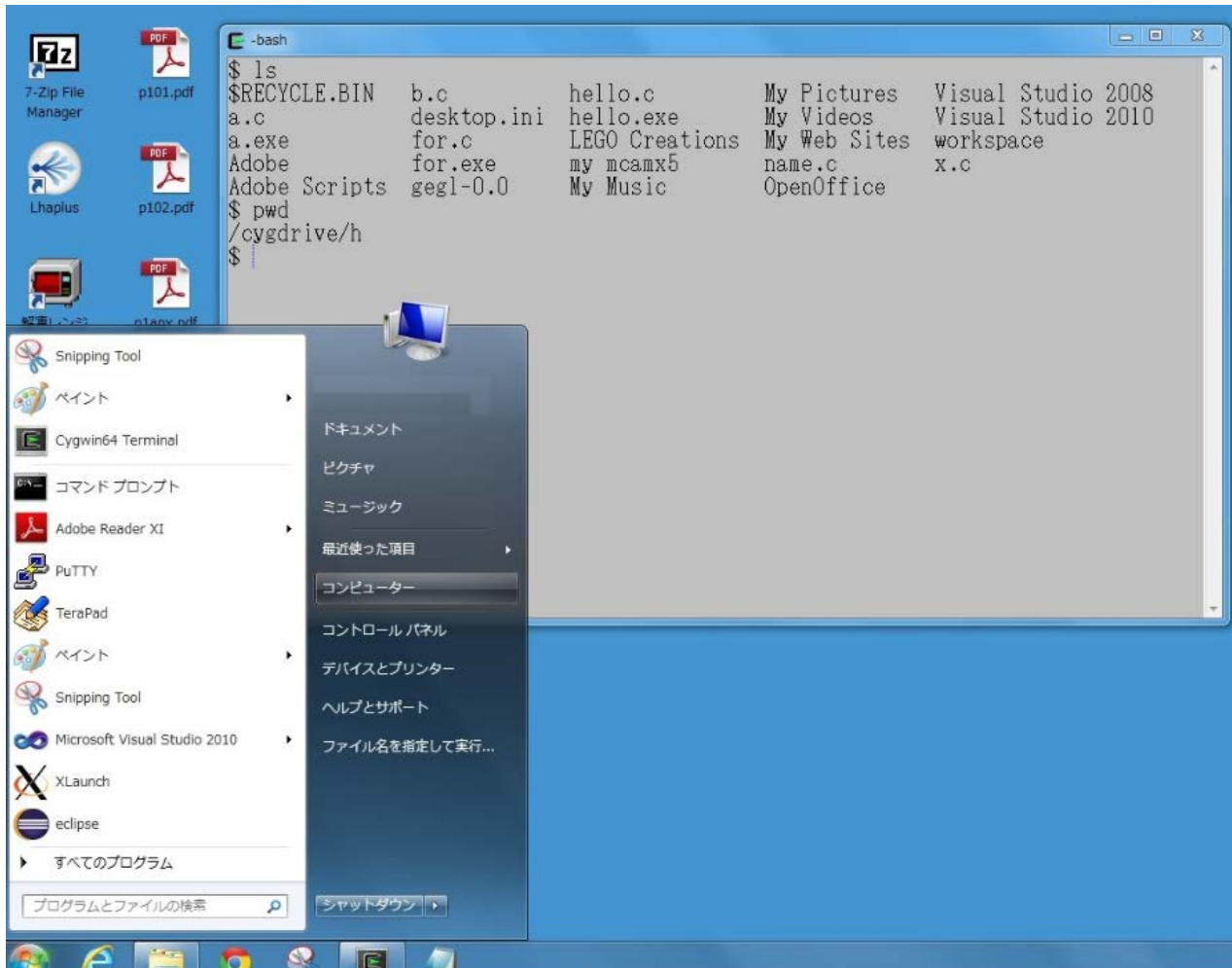


ファイルの在り処について

- 作成したソースプログラムや実行ファイル等は、通常、ネットワークの場所(Hドライブ)に保存されている。
- 大学のCygwinがそういう設定のようだ。
- 保存されているフォルダにたどり着く手順を以降述べる。
- もし、デスクトップ等にファイルを作成してしまった場合は、下記に従いCygwinの保存場所にコピーするように。

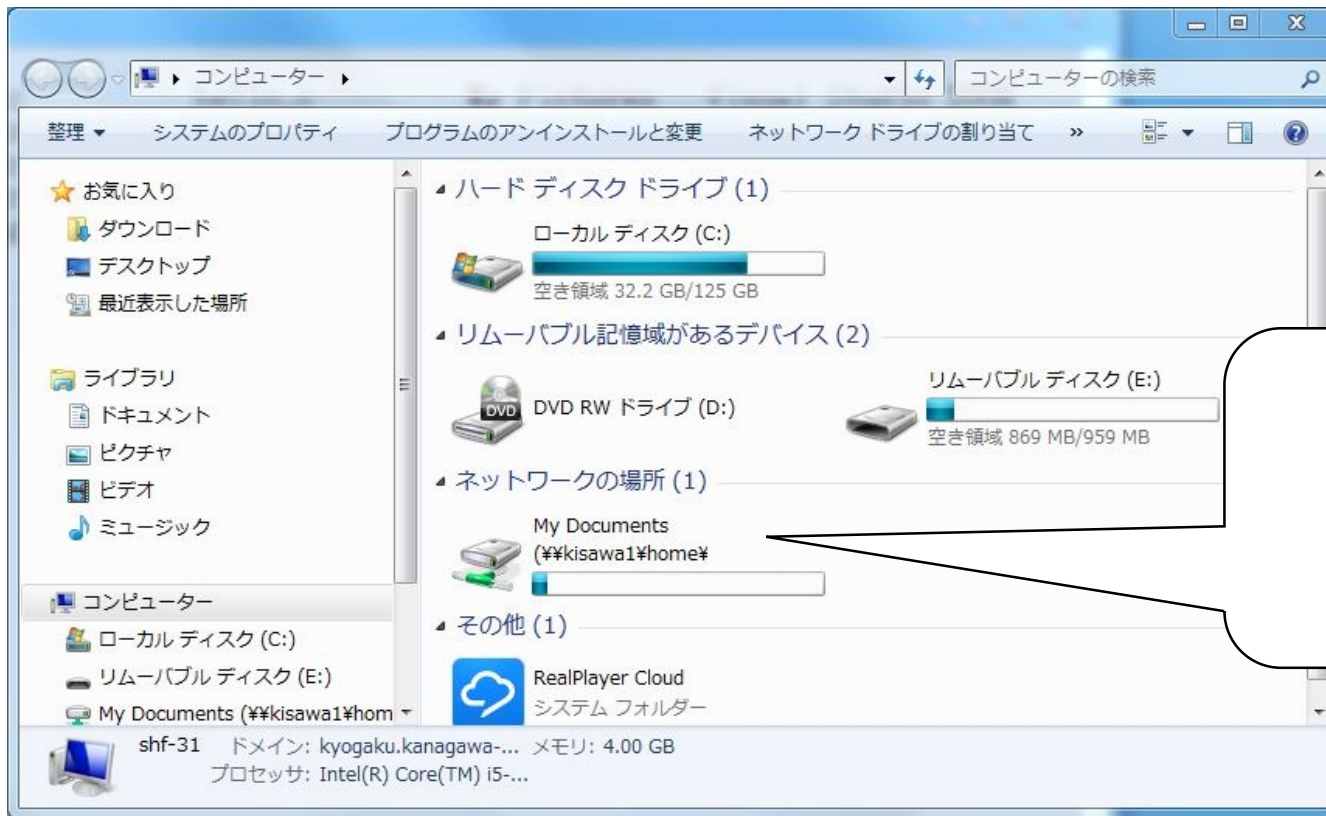
ステップ 1/3

- スタートメニューからコンピュータを選択する.



ステップ 2/3

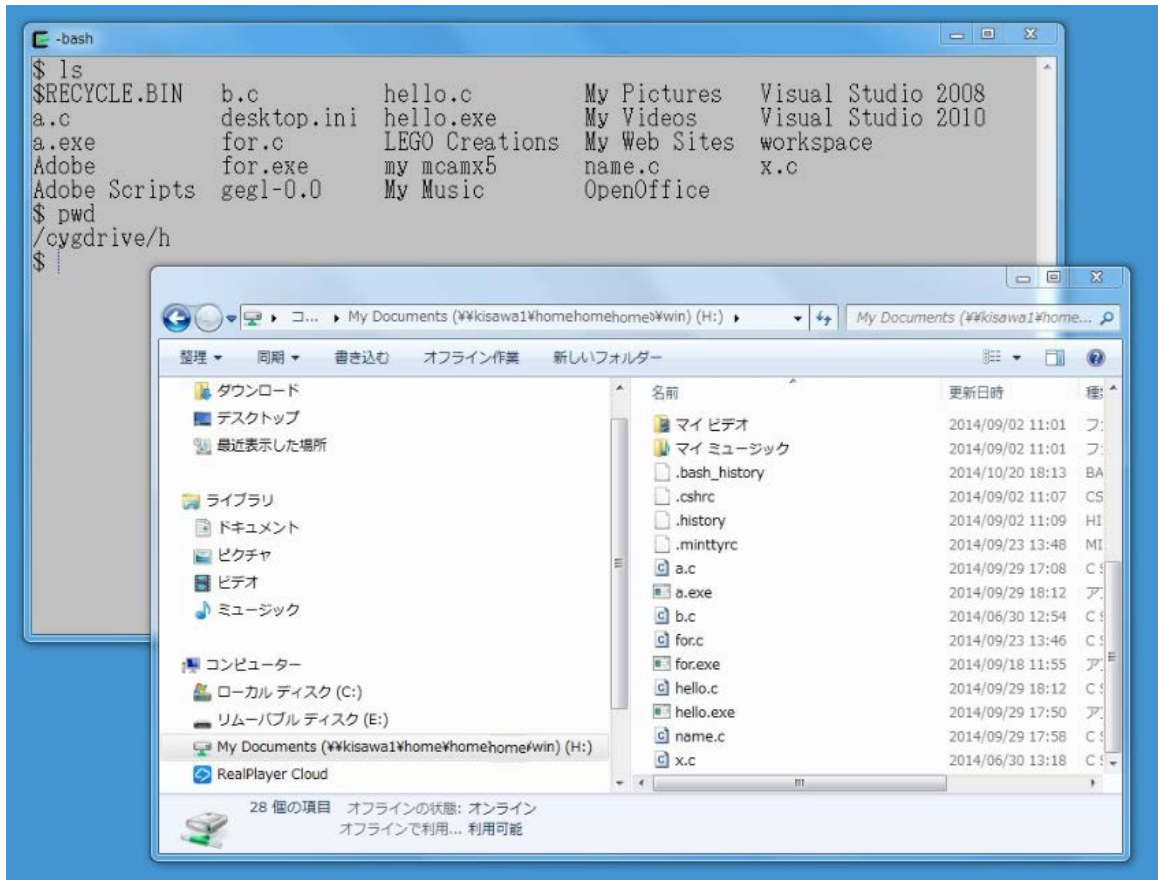
- ネットワークドライブをクリック



コレ

ステップ 3/3

- 以下の感じでなんとか.c 等が見える.
- 他にファイルを作ってしまった場合, ココにコピーする.



停止プロセスの削除について

- cc コマンドで, a.exe が作成できないというエラーが出る場合があります.

```
$ cc circle.c  
/usr/lib/gcc/x86_64-pc-cygwin/4.9.2/../../../../x86_64-pc-cygwin/bin/ld: 出力フ  
ァイル a.exe が開けません: Device or resource busy  
collect2: エラー: ld はステータス 1 で終了しました
```

- これは, a.exe ファイルが利用中である場合におきます.
- 多くは, 実行中の a.exe が休止していることが原因です.
 - 多分, コントロールZ 等を押したせいだと思う.
- 休止している a.exe を以下のステップで, 全て停止させてから, 改めて, cc コマンドを実行してください.

休止プロセスの除去

1. notepad 等, テキストエディタを全て終了する.
2. jobs というコマンドで, 休止中のプログラムが表示されるか確認.
3. 確認された場合, fg というコマンドを実行.
 1. これで, 休止プログラムの一つの実行が再開される.
4. コントロールC を押して, 休止から復旧したプログラムを強制的に殺す.
5. jobs で何も表示されなくなるまで, 2から4を繰り返す.

画面例

```
ft101932ao@shf-71 ~
$ cc circle.c
/usr/lib/gcc/x86_64-pc-cygwin/4.9.2/../../../../x86_64-pc-cygwin/bin/ld: 出力フ
ァイル a.exe が開けません: Device or resource busy
collect2: エラー: ld はステータス 1 で終了しました

ft101932ao@shf-71 ~
$ jobs
[1]  停止                ./a
[2]- 停止                ./a
[3]+ 停止                ./a

ft101932ao@shf-71 ~
$ fg
./a

ft101932ao@shf-71 ~
$ jobs
[1]- 停止                ./a
[2]+ 停止                ./a

ft101932ao@shf-71 ~
$
```

ここでコン
ロールCを
押す (繰返)

```
ft101932ao@shf-71 ~
$ jobs
[1]+ 停止                ./a

ft101932ao@shf-71 ~
$ fg
./a

ft101932ao@shf-71 ~
$ jobs

ft101932ao@shf-71 ~
$ cc circle.c

ft101932ao@shf-71 ~
$
```

前述のステップとの対応

プログラミング

手作業
エディタ

ステップ
4,5



コンパイル

自動変換
コンパイラを使用

ステップ
6



実行

生成された
実行ファイルを使用

ステップ
7



開発ツールの
準備等

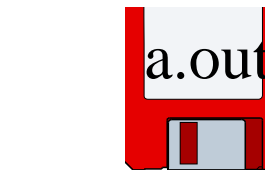
ステップ
1,2,3



ソースプログラム
(ソースコード
hoge.c 等)

レポート提出と
後始末

ステップ
8,9



実行ファイル
(ロードモジュール
a.out 等)

その他, TIPS

「参考」とある部分は
若干高度な内容

ファイルの存在確認 ls

- ls という名前のコマンドで、今あるファイルの名前一覧が確認できる。
- もし、自分が作成した nantoka.c 等のファイルが見当たらない場合、作成ミスがあることが分かる。

```
コマンドプロンプト - bash
$ ls
TRASH a.bat a.exe nantoka.c
$
```

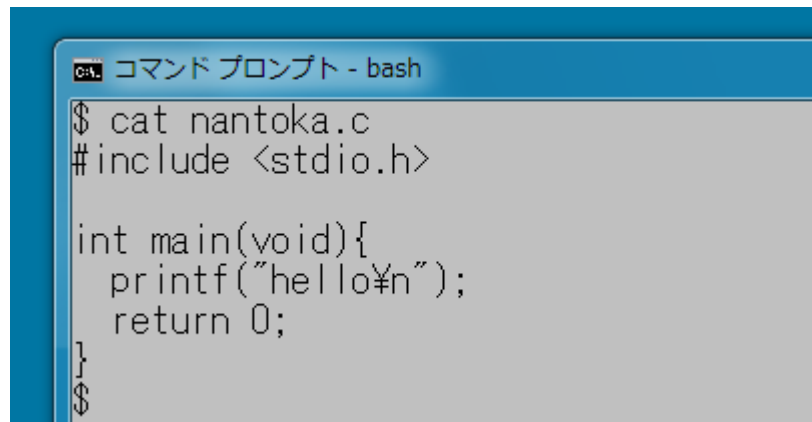
参考:

a.exe 等の実行ファイルは、かなりサイズが大きいことも分かる。

```
コマンドプロンプト - bash
$ ls -l
total 69
drwxr-xr-x 2 kaiya Administrators 0 Sep 26 12:57 TRASH
-rwxr-xr-x 1 kaiya Administrators 82 Dec 15 2015 a.bat
-rwxr-xr-x 1 kaiya Administrators 68398 Sep 27 15:35 a.exe
-rw-r--r-- 1 kaiya Administrators 77 Sep 27 15:34 nantoka.c
$
```

ファイルの中身の確認 cat

- cat というコマンドで、ファイルの中身を確認できる.
- ただし, nantoka.c 等のソースファイルの中身のみを確認すること.
- a.exe 等の実行ファイルの中身を確認してはいけない.
 - 中はマシン語で、画面に表示できない文字(データ)を多数含むため.



```
cmd コマンドプロンプト - bash
$ cat nantoka.c
#include <stdio.h>

int main(void){
    printf("hello\n");
    return 0;
}
```


参考 cc -o の意味

- ccコマンドは, nantoka.c を実行ファイル(マシン語)に変換する.
- 実行ファイルは Windows では, なんとか.exe という名前になっている.
- ccコマンドは, 明示的に指定しないと, a.exe という名前の実行ファイルを作成する.
- もし, 実行ファイルの名前を指定したければ, -o のあとに, ファイル名 (.exeを除く)を指定する.

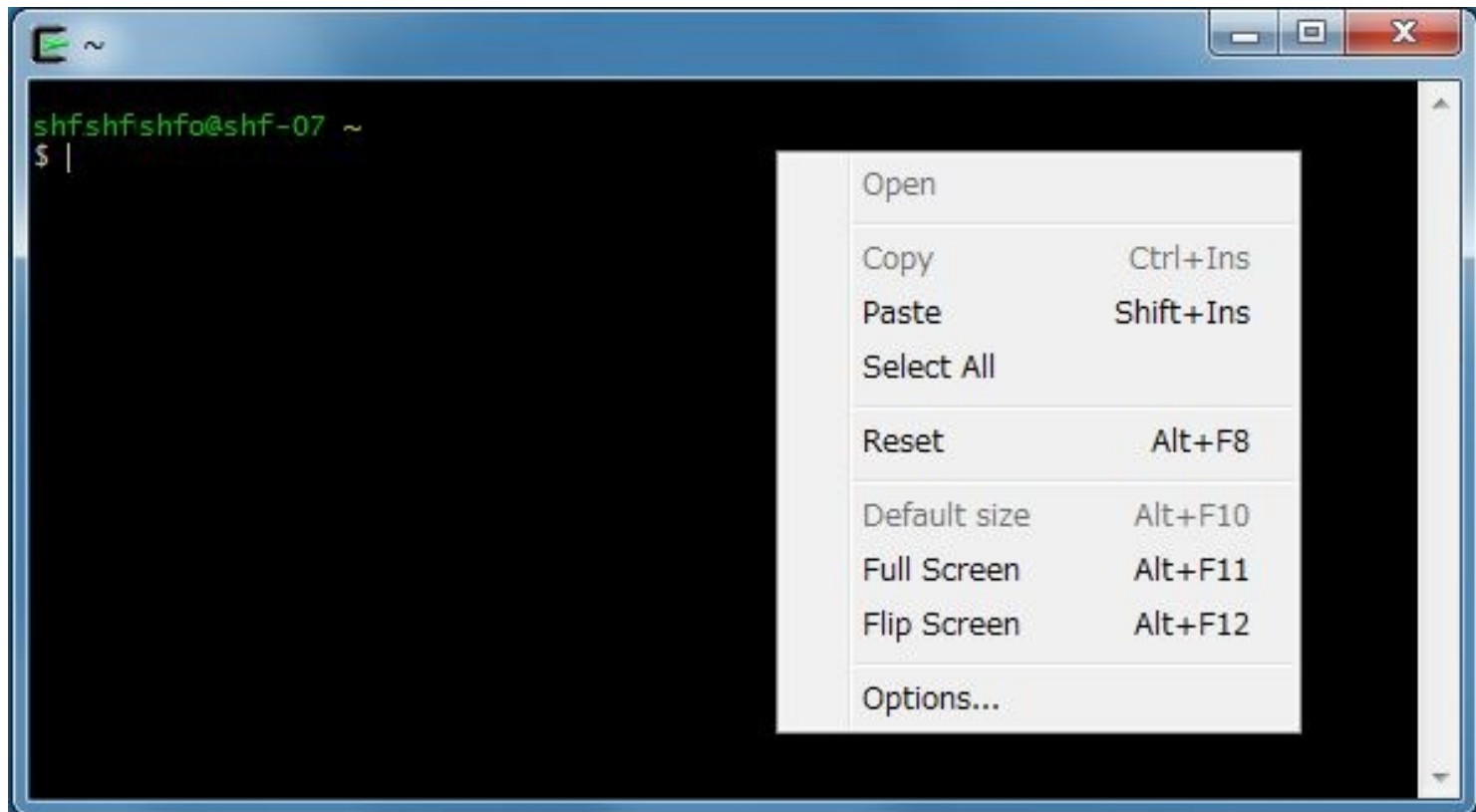
```
コマンド プロンプト - sh
$ ls
TRASH a.bat hello.c nantoka.c sample02
$ cc hello.c
$ ls
TRASH a.bat a.exe hello.c nantoka.c sample02
$ ./a.exe
hello
$ cc -o sayhello hello.c
$ ls
TRASH a.bat a.exe hello.c nantoka.c sample02 sayhello.exe
$ ./sayhello.exe
hello
$
```

参考 . / とは何か？

- 現在のフォルダ(ディレクトリ)にある a.exe 等の実行ファイルを実行する際には, 単に a.exe と書くだけではダメ.
- ./a.exe もしくは ./a と書かなければならない.
- . は多くのOSで現在のフォルダを意味する.
- / は, UNIX系OSであるフォルダの下を意味する.
 - Windows系の場合, ¥ もしくは \ を使う.
- よって, ./ は, 「現在のフォルダの下にある」を意味する.
- セキュリティの関係で, 多くのUNIX系OSは, 現在のフォルダ直下の実行プログラムを指定するために, ./ をつけなければならないようになっている.

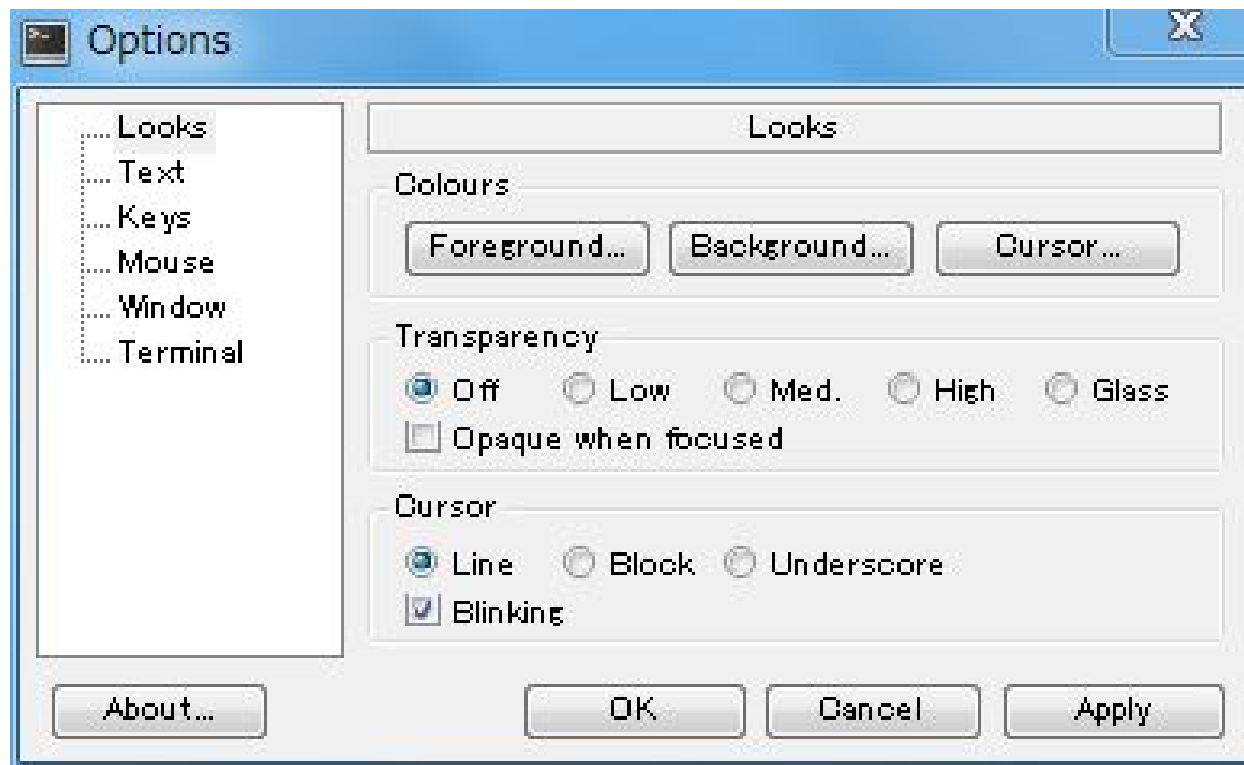
Cygwinでの日本語表示設定 1/3

- 警告等の日本語を表示したい場合，以下の手順で，設定を行います．
- まず，右クリックでメニューリストを出して，[Options]を選択する．



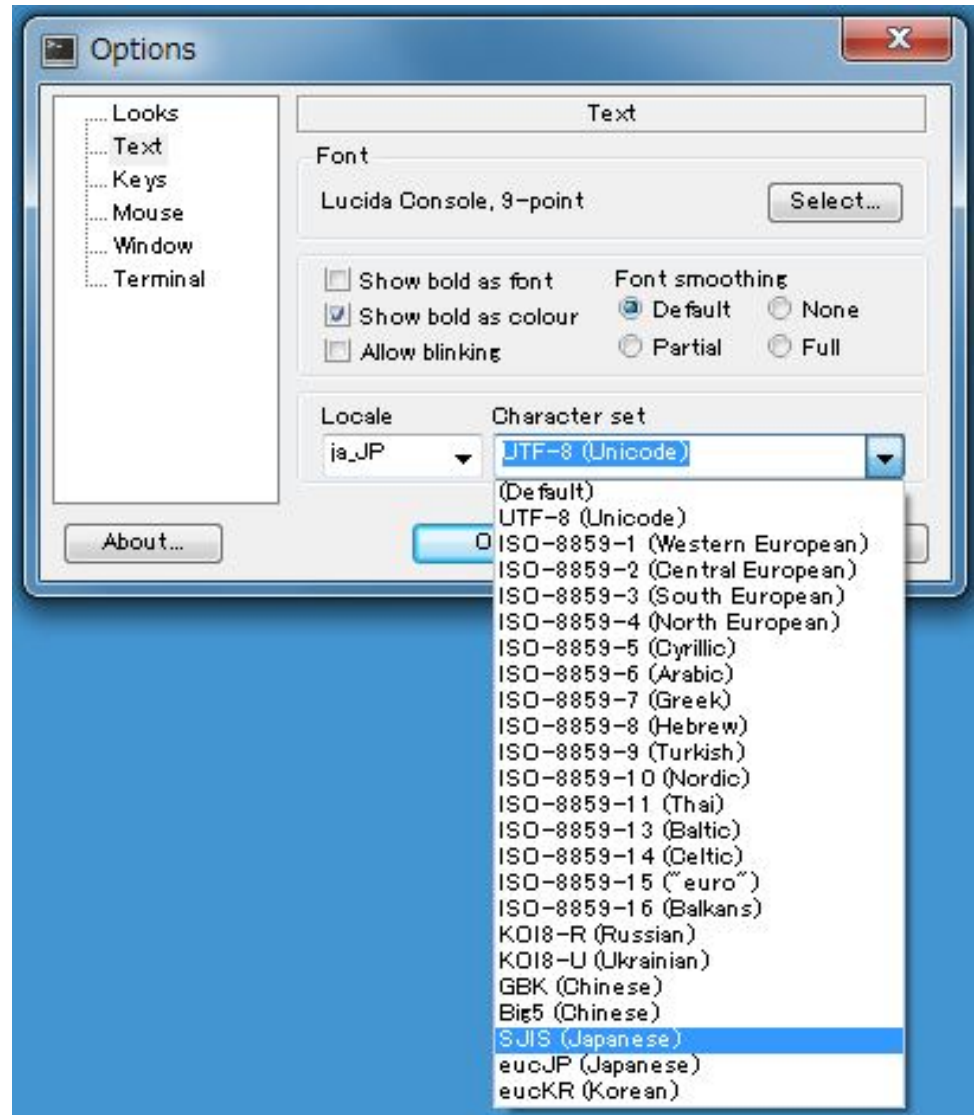
Cygwinでの日本語表示設定 2/3

- Options画面で[Text]を選択



Cygwinでの日本語表示設定 3/3

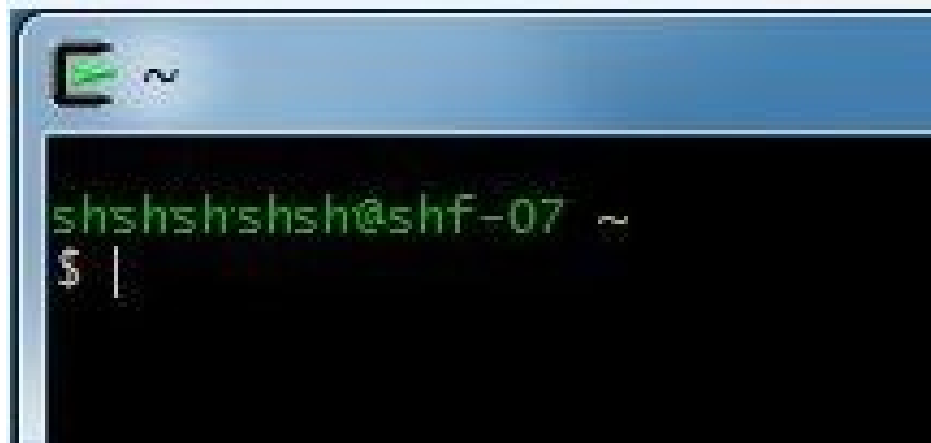
- Locale と Character set という値をそれぞれ以下に設定.
- Locale ja_JP
- Character set は, UTF-8 (Unicode) を選択.



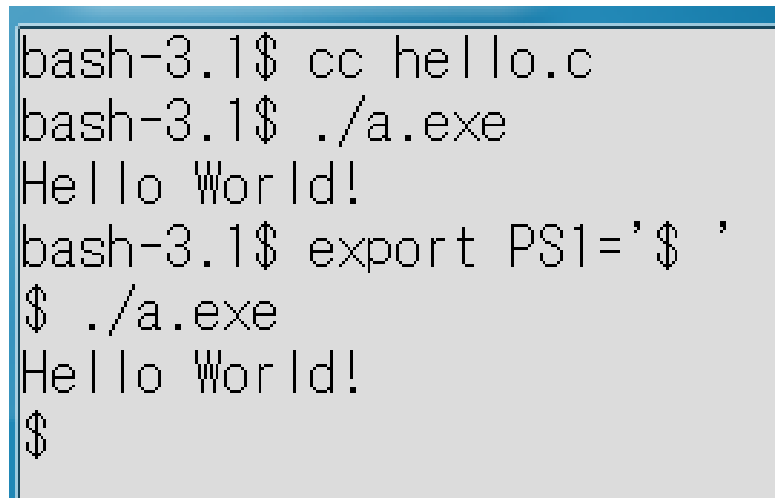
参考 プロンプト

- Terminalが人間の入力待ち状態である際に出てる短い文字を「プロンプト」といいます.
- Cygwinのプロンプトは若干長いです. (左)
- 邪魔な場合, 短くすることができます. (右)

`export PS1=' $ '` という命令



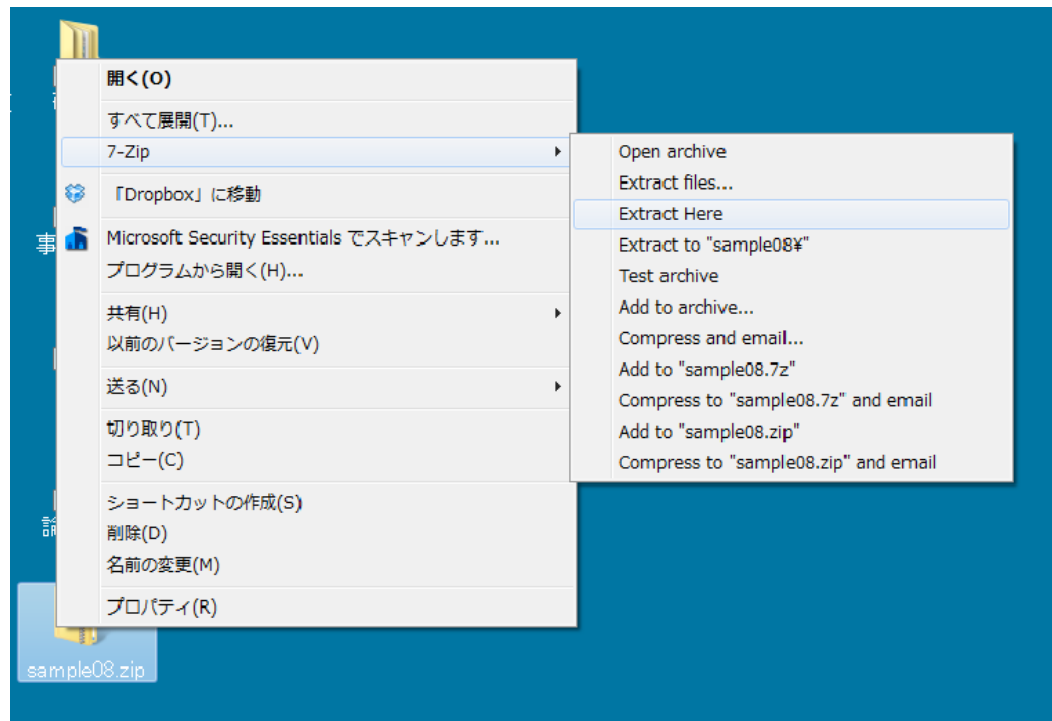
```
shshshshsh@shf-07 ~  
$ |
```



```
bash-3.1$ cc hello.c  
bash-3.1$ ./a.exe  
Hello World!  
bash-3.1$ export PS1=' $ '  
$ ./a.exe  
Hello World!  
$
```

Zipアーカイブについて

- 複数のファイルを配布する際、アーカイブファイルという1個のファイルにまとめることが多い。
- Zipはその中でも、ポピュラーなものの一つである。
- 本授業ではサンプル等をZipで配布する機会が多い。



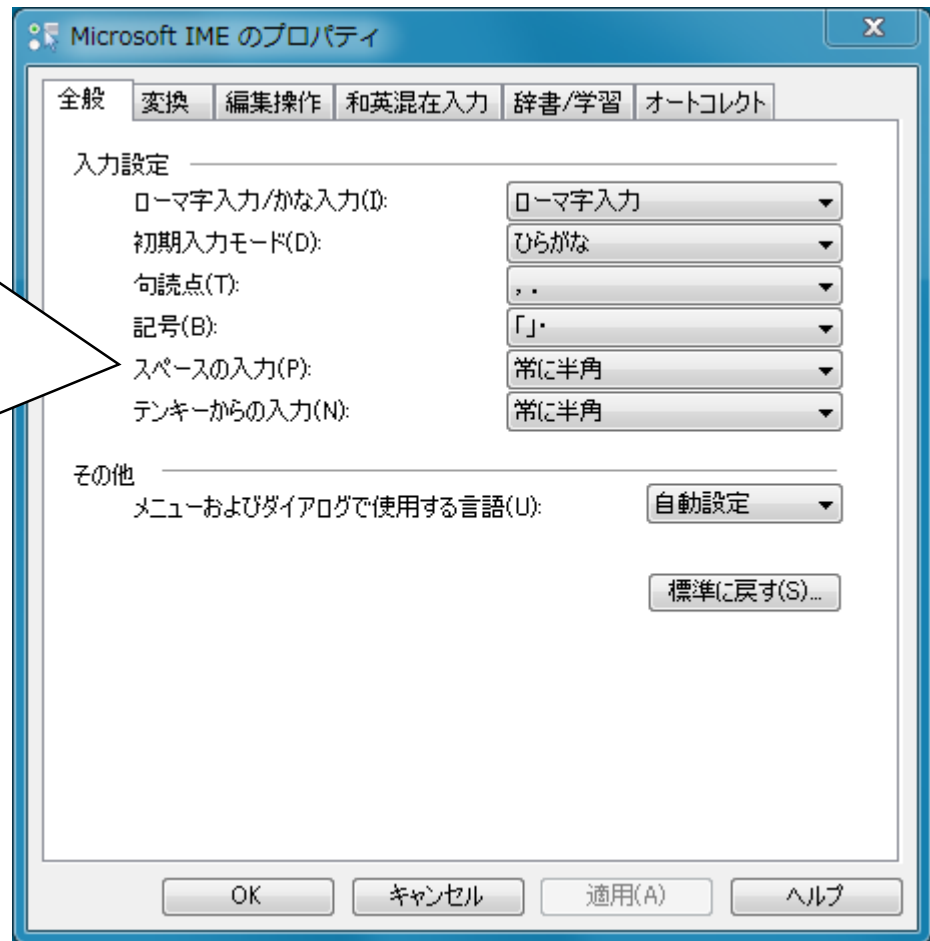
本授業でのZipの使い方

1. 授業等のページからブラウザでダウンロードする.
2. 通常 ¥Users¥自分のユーザー名¥Downloads¥ というフォルダにダウンロードされる.
3. ダウンロードした `nantoka.zip` を, `H:¥` に移動する.
4. エクスプローラーで当該ファイルを選択し, 右クリックでメニューを出し, 7-Zip ⇒ Extract Here を選択.
5. 普段, `cc` 等のコマンドを実行するフォルダに, 例えば, `sample08¥` というフォルダが作成され, その中にサンプルプログラム等が展開される.
 - 直接, ファイルが展開される場合もある.
6. サンプルを見たり実行したりしたい場合は, `cd sample08` 等のコマンドで, フォルダを移動してから, `cc` や `notepad` を利用する

参考 全角スペースは無効化がベター

少なくとも日本語の空白文字はプログラムでは害悪しかないので、無効化するのがよいと思います。

まあ、氏名の区切りとかで全角スペースを強要する残念なサイトとかもありますが、orz



以上