プログラミングI 第1回
数理物理，総合理学等向け

2018年9月24日

海谷 治彦

目次

教科書 第1章 表示の内容となります.

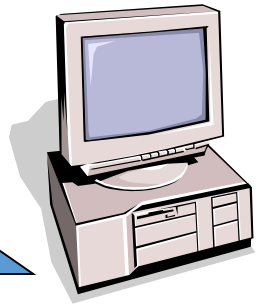
- 開発の流れの復習
- 前回の補足
  - サンプルの動かし方等
- 基本的なプログラムの構造
- 関数
- 文字列を表示する関数 printf
- ¥n の使い方
- include文について

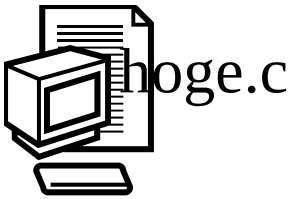
復習: Cプログラムの開発の流れ

プログラミング
手作業
エディタを使用

コンパイル
自動変換
コンパイラを使用

実行
生成された
実行ファイルを使用





ソースプログラム
(ソースコード
hoge.c 等)

実行ファイル
(ロードモジュール
a.out 等)

いくつかの用語確認

- ソースプログラム
  - **Source Program**
  - Source Code (ソースコード)とも呼ばれる.
  - C言語の文法に則して記述されたコンピュータへの命令が記述されたファイル
- 実行ファイル
  - **Load Module** (ロードモジュール)とも呼ばれる.
  - 実行可能プログラムとも呼ばれる.
  - ファイルの中身はコンピュータの言葉(マシン語)による命令内容他が書いてある.
    - 人間が読んでも意味がわからない.
  - Terapad.exe 等のファイルのこと.
  - 特にファイル名を指定しなければ, a.exe という名前になる. (本物のUNIX系OSの場合は, a.out)

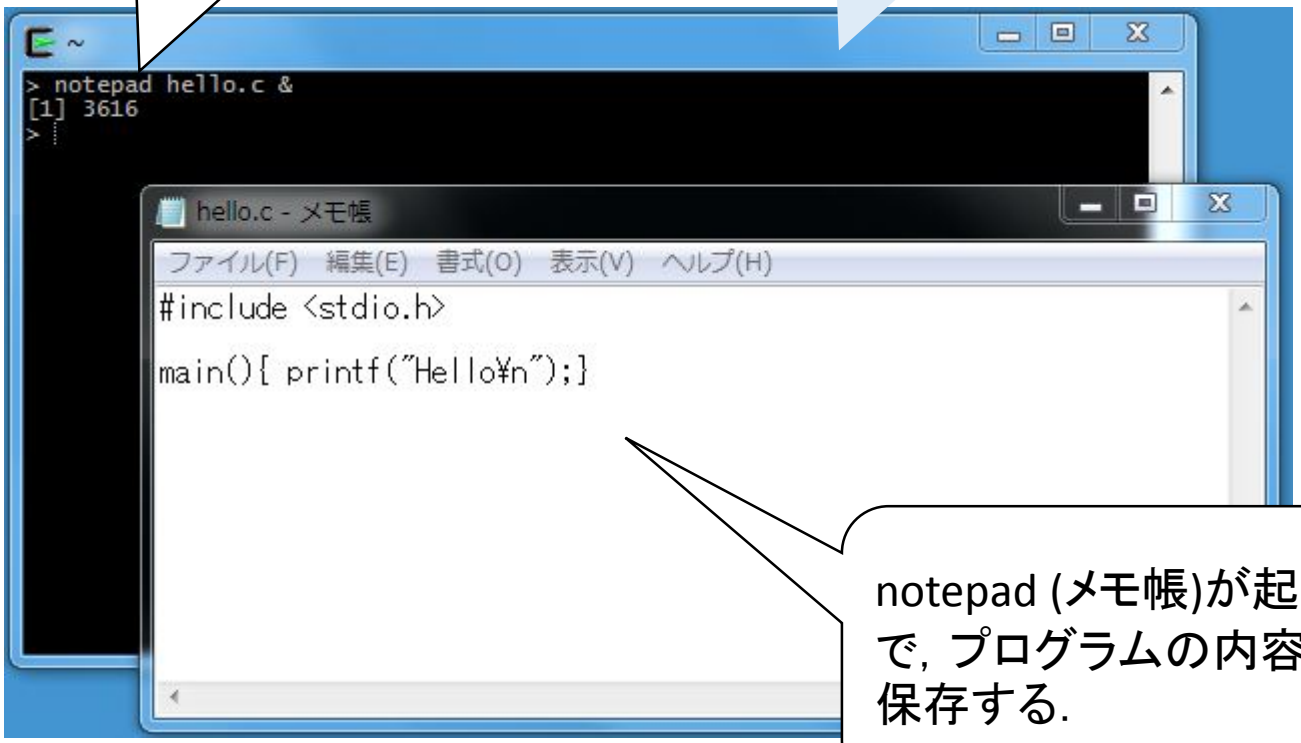
Cygwin64 Terminal

- 一般に「shell」とか「コマンド インタプリタ」とか言われるプログラム.
- プログラム名を打ち込んで, [Enter]キーを押すと, 当該のプログラムが実行される.
- 本授業で主に使うプログラム
  - notepad
    - Windowsのメモ帳
    - ソースプログラムのファイルを作成するのに使う.
    - ファイル名は「nantoka.c」等, 拡張子を .c とする.
  - cc
    - C Compiler
    - コンパイラ.
    - ソースプログラムを実行ファイルに変換するプログラム.
    - ソースプログラムに誤りがあると, その旨を表示する.

notepadの起動

notepad hello.c &
と入力して,
Enterキーを押す.

コレがCygwin64 Terminal



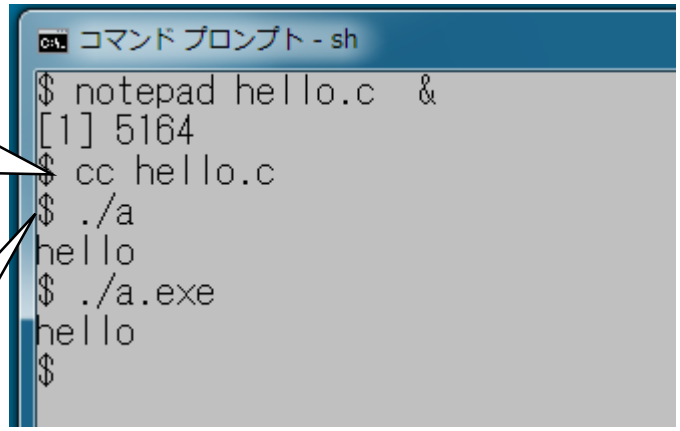
notepad (メモ帳)が起動するので, プログラムの内容をうって, 保存する.

コンパイラ ccの起動と実行プログラムの実行

cc hello.c
と入力して,
Enterキーを押す.

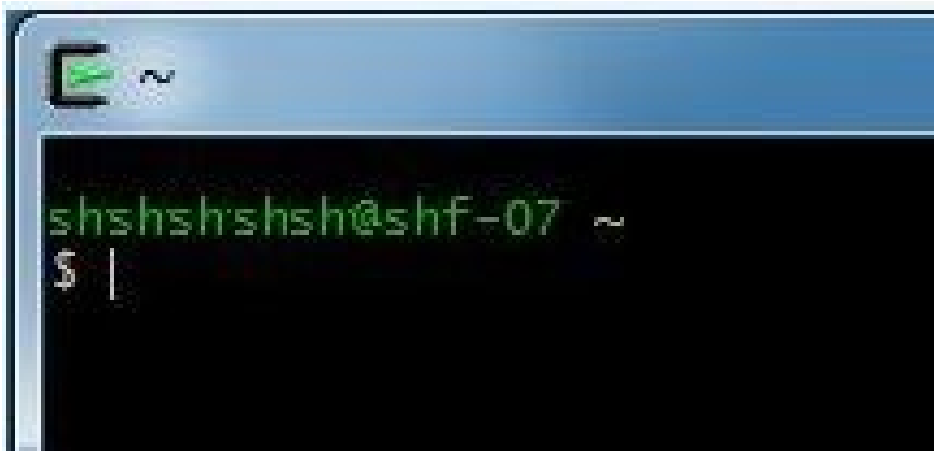
コンパイルが成功すると,
特に何も表示されない.

作成された実行ファイル a.exe を実行する. 実行は,
./a
とする. ./a.exe でもよい.
結果として, notepadで自分が書いたプログラムがコンピュータによって実行される.



プロンプト

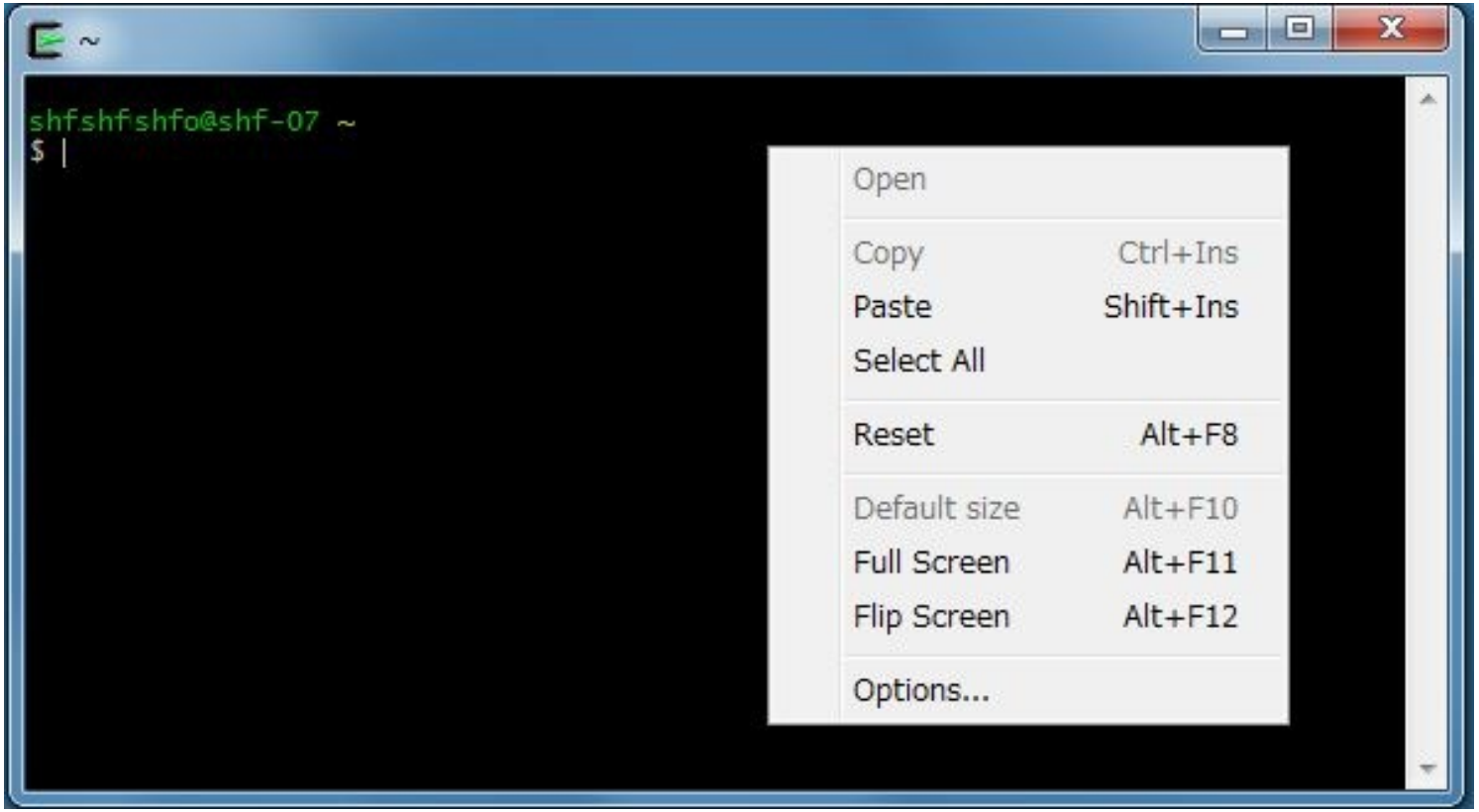
- Terminalが人間の入力待ち状態である際に出てる短い文字を「プロンプト」といいます.
- Cygwinのプロンプトは若干長いです. (左)
- 邪魔な場合, 短くすることができます. (右)




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

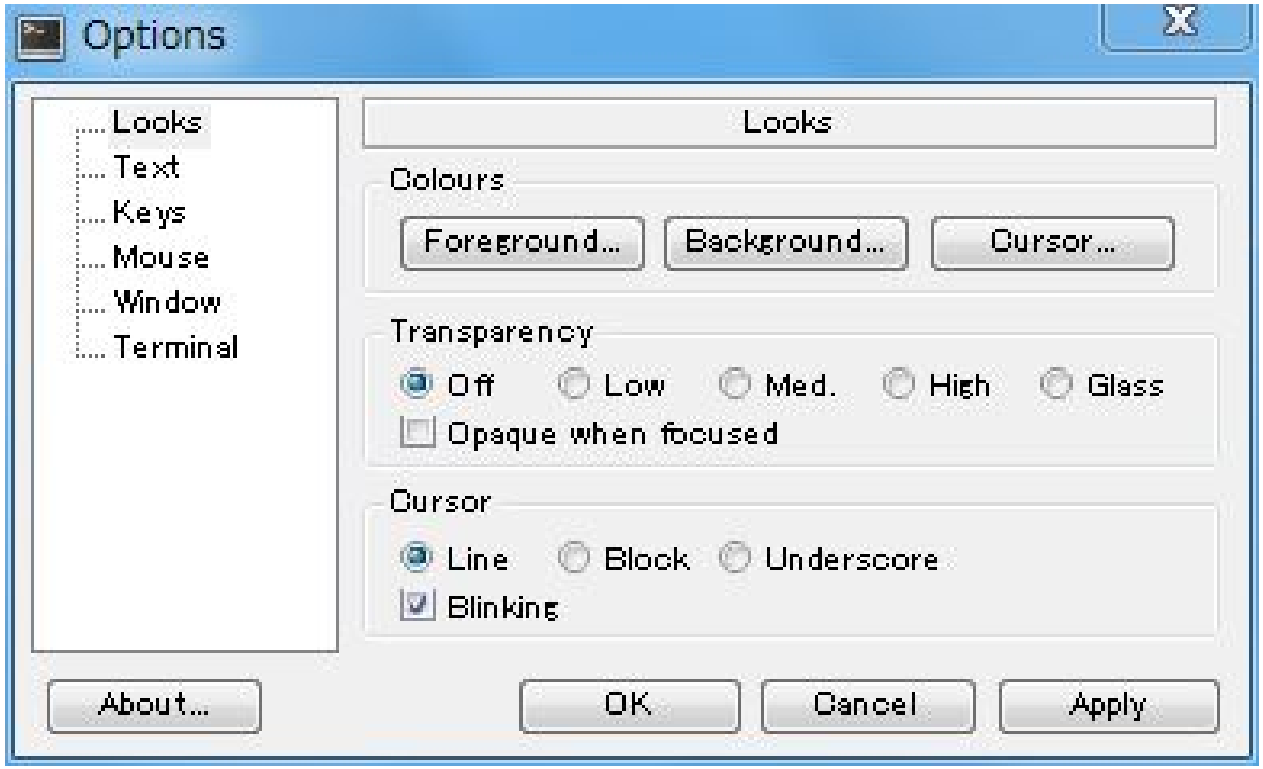
Cygwinでの日本語表示設定 1/3

- 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．



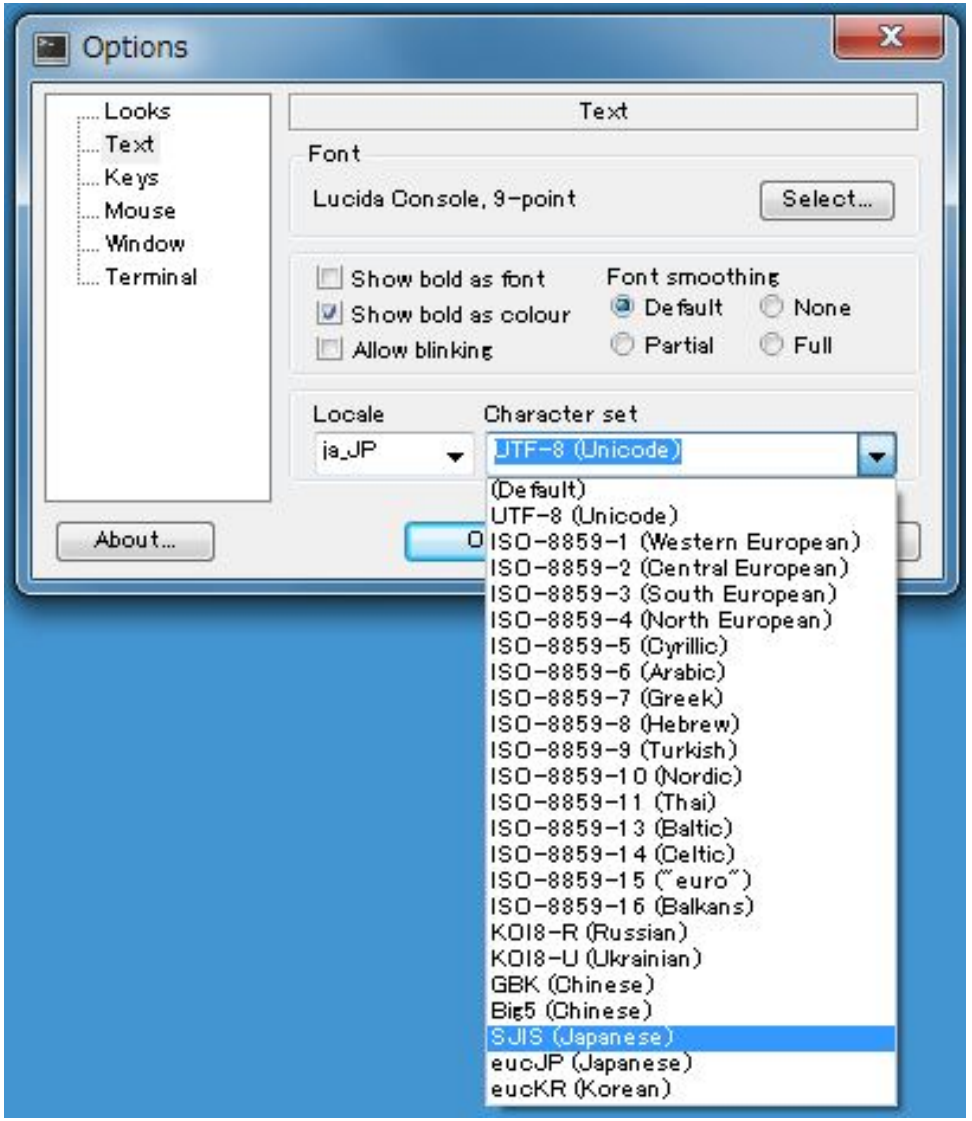
Cygwinでの日本語表示設定 2/3

- 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.



画面に文字列を表示 (レp.16 明p.8)

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world¥n");
    return(0);
}
```

教科書とちょっと違う

説明の都合，行番号つけます

```
1: #include <stdio.h>
2:
3: int main(void)
4: {
5:     printf("Hello, world¥n");
6:     return(0);
7: }
```

※ 注意！
行番号がついたままでは，コンパイルできません．

一行目，おまじない

- 一行目はあまり考えずに必ず書いてください.

- 詳細な意味は後日説明しますが，printfを使うための準備みたいなものです.
- 参考: Standard Input and Output header の略に相当します.

```
1: #include <stdio.h>
2:
.....
```

実行される命令群

- 4行目から7行目に実際に実行される命令を書きます.

```
2:
3: int main(void)
4: {
     中略
7: }
```

3〜7行目: 関数定義

- 3〜7行目は main という名前の「関数」を定義してます.
- 関数はひとまとまりの処理をグループ化したものです.
- 数学でやった関数とは若干違う部分もありますが, まぁ, 似たようなものです.

```
3: int main(void)
4: {
     中略
7: }
```

main 関数

- C言語では，必ず，main 関数 が無ければいけません.
- プログラムは必ず，main 関数から開始されます.

```
3: int main(void)
4: {
5:     printf("Hello, world¥n");
6:     return(0);
7: }
```

5行目 関数呼び出し

- C言語では他人様が作った関数を呼び出すことができます.
- 5行目では, printf という誰かが作った関数を呼び出しています.
- この関数が, どんな内容なのかは後述.

```
3: int main(void)
4: {
5:     printf("Hello, world¥n");
6:     return(0);
7: }
```

6行目 return (0)

- 数学の関数同様，Cの関数も返り値があります．
  - $f(x)=x^2+5$ なら，f(4)は返り値として21を返す等．
- この6行目は，関数mainが，常時，0 ゼロを返すということを示しています．
- main関数の返り値は，ちょっと特別で，実行ファイルを実行した環境(通常はshell)にその値が渡されます．
  - 慣例として，0は実行成功，それ以外は失敗を示すという意味づけがされています．

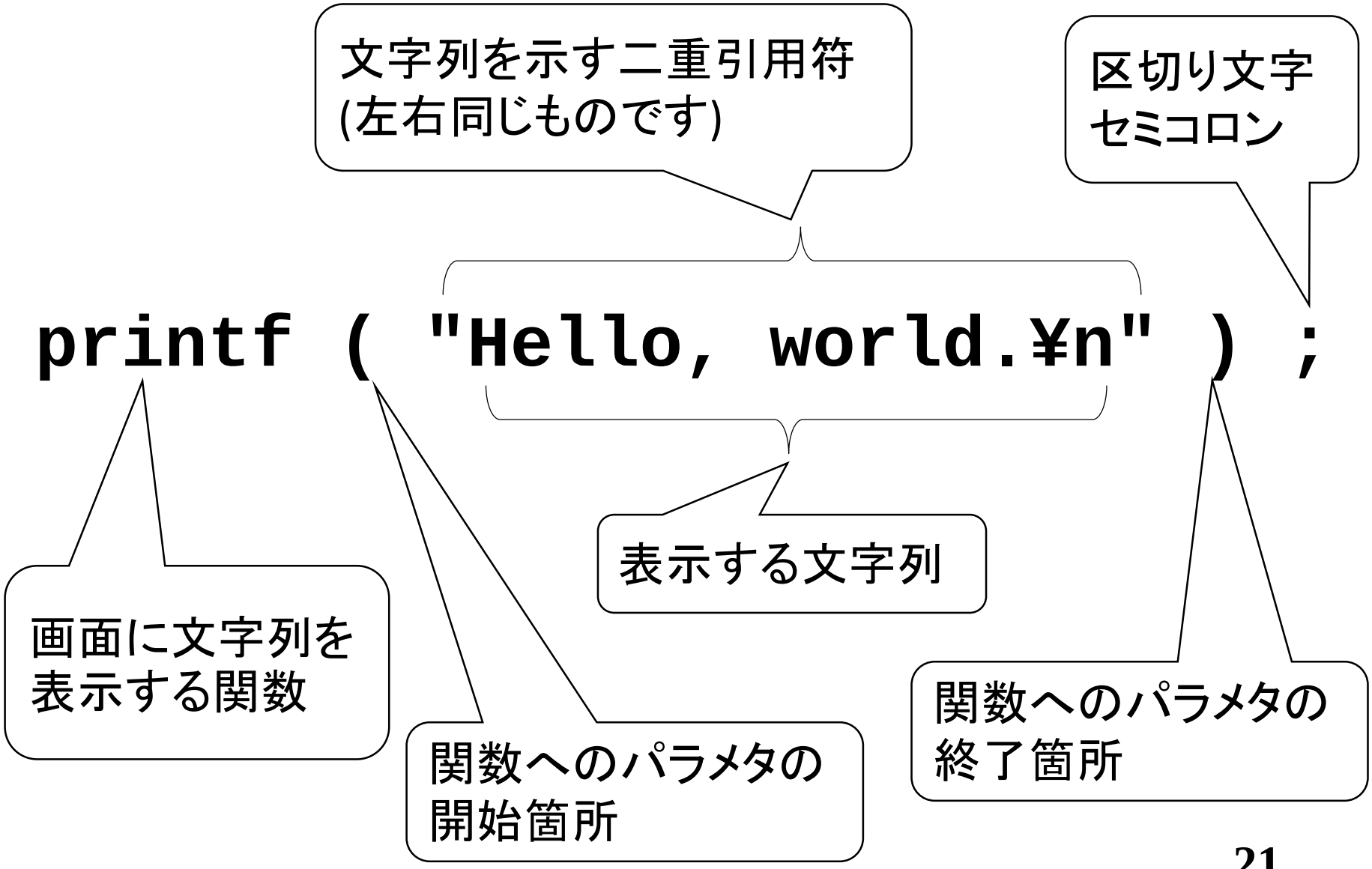
```
3: int main(void)
4: {
5:     printf("Hello, world¥n");
6:     return(0);
7: }
```

関数 printf (p.19〜)

```
printf("Hello, world¥n");
```

- 画面に文字列を表示する機能である.
- 基本，数学の f(x)=… 等の関数定義に対して，f(10) や f(21) 等を計算する「関数の利用」と同じである.
- 数学の関数と違うのは，
  - パラメータが数字以外でもよい
  - 返り値が無くてもよい (実際は有るが使ってない)
  - 最後に ; セミコロンがある. (文書の句点。に相当)
  - 副作用がある (画面に文字が表示される)

printfの分解図 レFig 1-3 p.19



結局，printfとは？

- `printf(文字列);`

- で，コンピュータに文字列を画面に表示せよ！ということを命令している.
  - 文字列については後述
- 最後の；セミコロンは命令の区切りであり，日本語の「。」に相当する.

文字列

- 二重引用符 " でくくられた文字の列.
- プログラムによって文字を表示する際に用いる.
  - printf の例が典型例
- プログラム中で文字を処理する場合にも用いる.
  - たとえばユーザー名とパスワード等
- 単純に表示できない特殊な文字を¥を用いて書く.
  - 例: 改行文字の ¥n 等.

```
"Hello World!¥n"
"Hello¥n World.¥n"
"Hello world¥n¥n"
```

練習問題

- 以下のプログラムを改造して，コンピュータに

  Kanagawa University

- という文字を表示するプログラムにせよ.

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world¥n");
    return(0);
}
```

printfは複数回，使ってよい

- 数学の関数と同様，printfという関数は繰り返し，複数回利用できる.
- 結果として，コンピュータに複数の文字列を表示するように命令できる.

```
#include <stdio.h>

int main(void)
{
    printf("Hello, world¥n");
    printf("Hello, Japan¥n");
    printf("Konnichiwa¥n");
    return(0);
}
```

¥nの意味の確認

- 以下の二つのプログラムは同じ結果を表示する.

```
#include <stdio.h>



int main(void)
{
    printf("Hello, world¥n");
    return(0);
}
```

```
#include <stdio.h>



int main(void)
{
    printf("Hello, ");
    printf("world¥n");
    return(0);
}
```

ccの警告 プログラムの誤り

- コンパイラ(cc)は，プログラム中の命令が，文法にあっていないと，その旨を警告します．
- この警告は誤り修正に役立つ場合があります．

```
$ gcc hello.c
hello.c: In function 'main':
hello.c:7:9: warning: missing terminating ″ character [enabled by default]
  printf(″Hello World!¥″)
         ^
hello.c:7:2: error: missing terminating ″ character
  printf(″Hello World!¥″)
  ^
hello.c:8:2: error: expected expression before 'return'
  return (0);
  ^
hello.c:9:1: error: expected ';' before '}' token
 }
 ^
```

```
#include <stdio.h>



int main(void)
{
    printf("Hello, world¥")
    return(0);
}
```

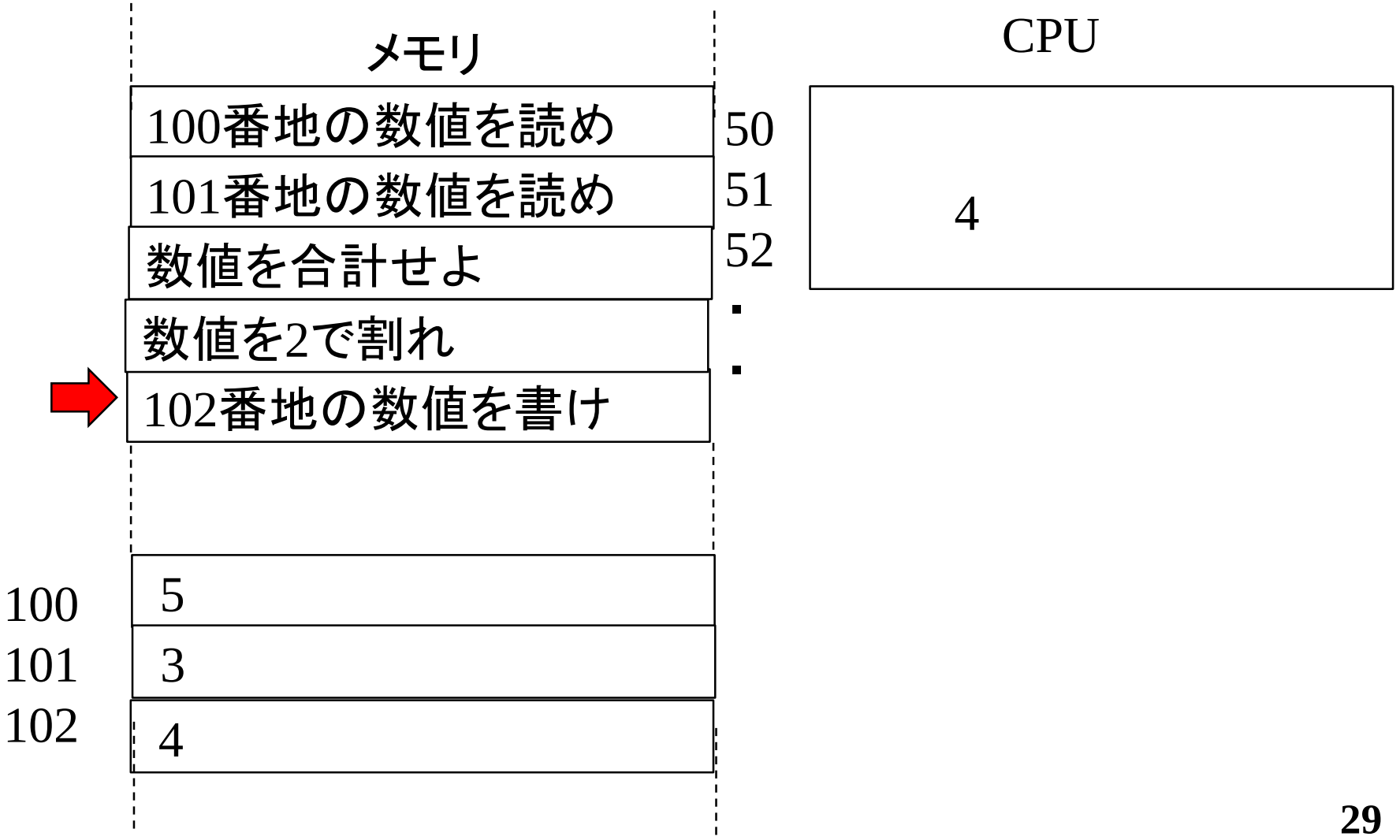
参考 stdio.h について

- #include <stdio.h>
- Standard Input and Output Header の略.
- この章のプログラムでは，printf を正しく使うために，この呪文が必要.
- 実際に，stdio.h というファイルが存在し，それをプログラムに包含(inlucde)させている.
  - 一般には，/usr/include/stdio.h にあるが，Cygwinは違うかも.
  - Linuxでは 900行ほどのファイル，わりと大きい.
- stdio.h ファイル中では，printf関数が，
  - どんな引数をとるか (文字列)
  - どんな返り値を返すか (使ってないが整数)

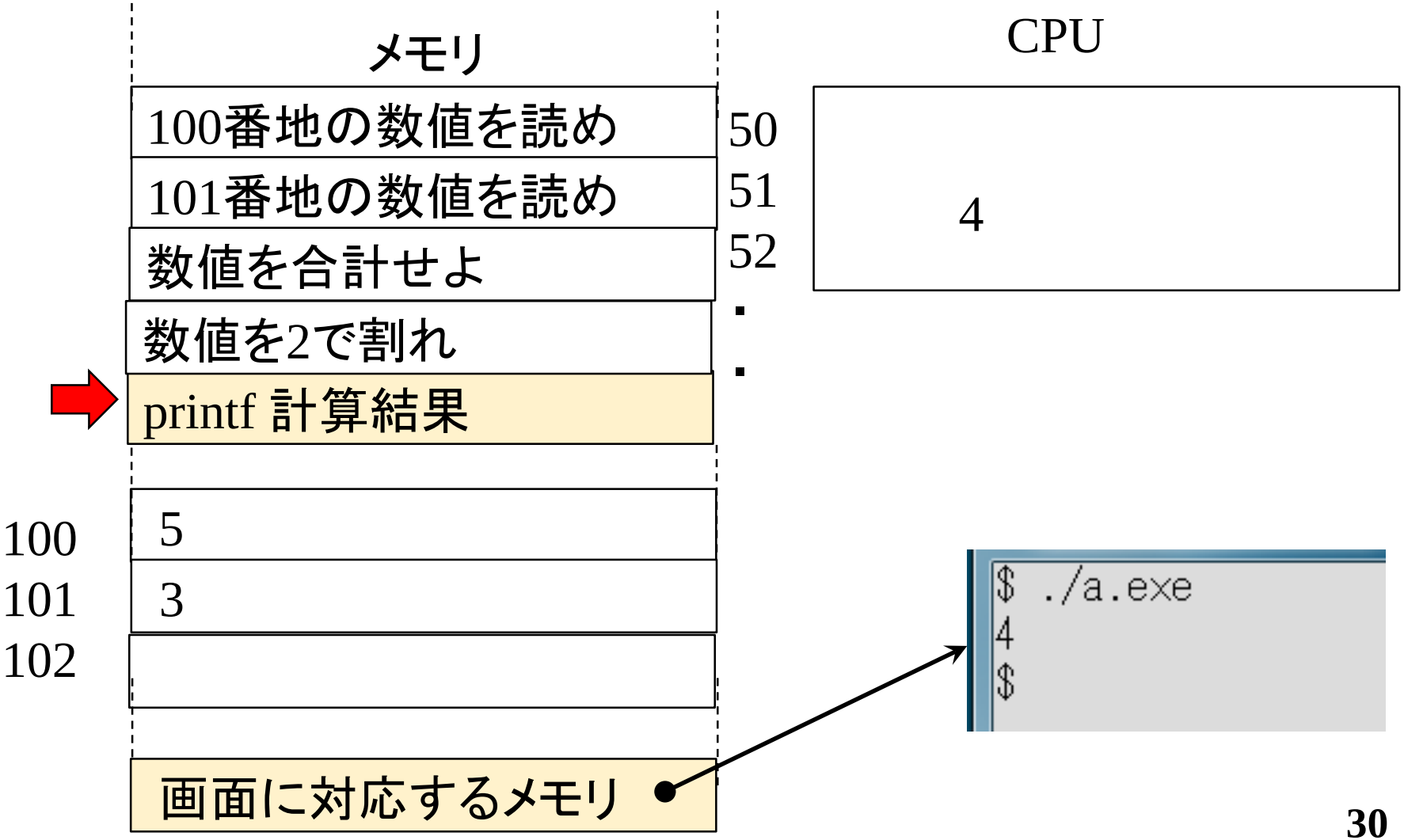
の定義がかかれている.

再掲載

簡易な例題 ～ 二値の平均

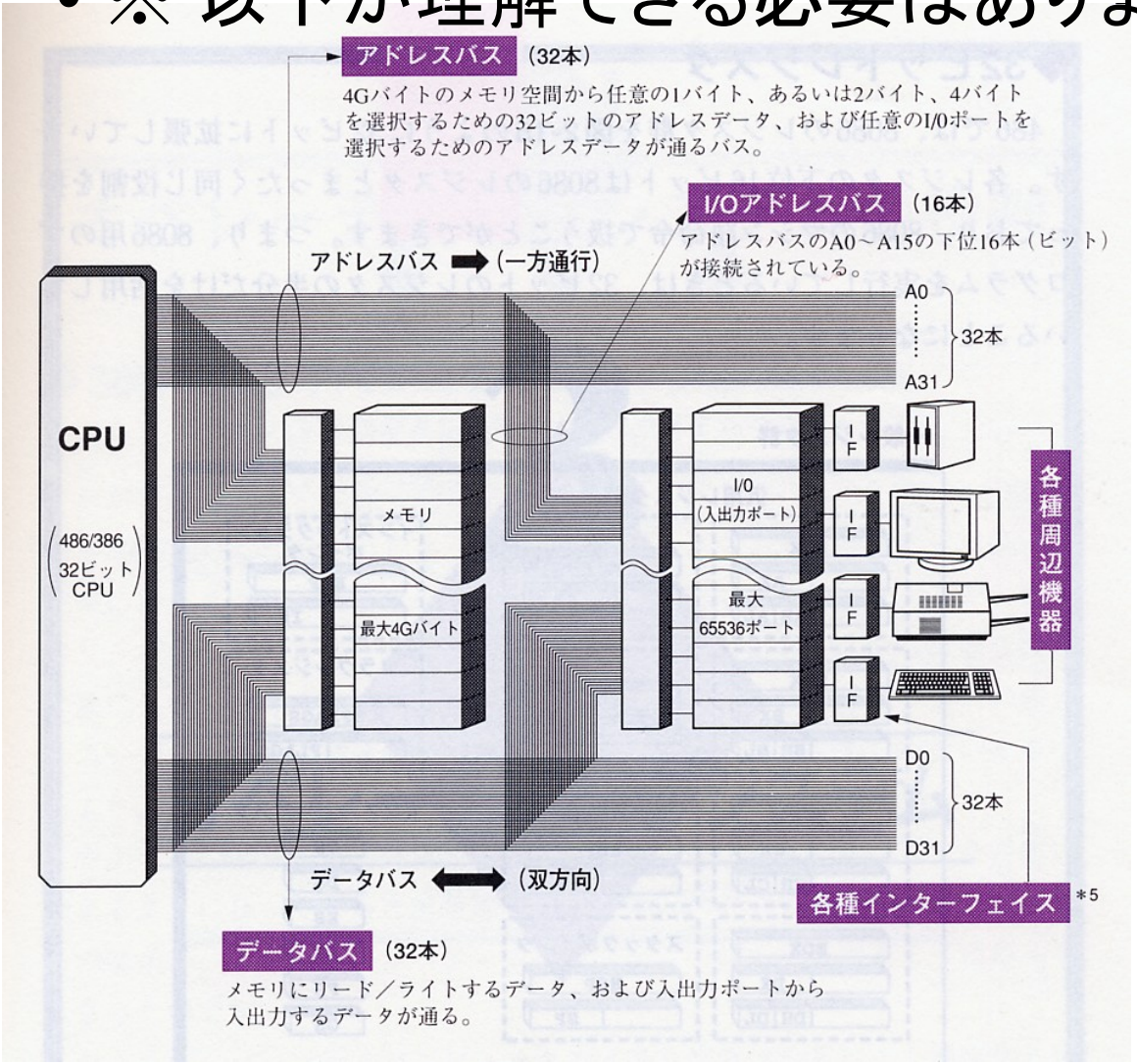


printfの動作イメージ



参考: よりホンモノなPCの構造

- ※ 以下が理解できる必要はありません. 参考です.



初めて読む486より

本日の演習 (演習1)

- 画面に自分の姓と名を行を分けて出力するようなプログラムを作成せよ. (以下の例参照)
- ソースプログラム名は name.c としてください.
- 次の授業までに dotcampus に出してください.

期待される結果の例

```
$ cc name.c
$ ./a
Yamada
Taro
$
```

本日は以上