

第3回

- 宿題のコメント まだの人は5月13日まで
- 電子メールの注意
- 情報の原理的事項(情報システムの原理と構成、各要素の役割等)
 - 二進法、bit、byte
 - 計算機の仕組み(ソフトウェア)
- C言語による初級プログラミング (1)
 - プログラムを作るとは
 - C言語でのプログラムの作成の手順
 - コマンドの使い方
 - C言語の大まかな構造
 - 変数と演算子の概要
 - Forによる繰り返しとIfによる分岐
 - (できれば)Whileによる繰り返しとSwitchによる分岐



宿題のコメント

■ 要約するパラグラフの取捨選択

- × : 各パラグラフから満遍なく要約。
- ○ : 大切なパラグラフ(複数も可)を要約。

■ 要約の方法

- × : キーセンテンス(大切な文章)を抜書き。
- △ : キーワードを抽出して組み立てる。
- ○ : キーワードを中心に新たに組み立てる。

(例えば:)

情報処理教育の目的は、単に情報機器を扱えるといったレベルに留まらず、情報社会からの的確なものを選別する能力と、急速に進化する情報技術に対応できる技能を獲得する機会を提供することである。

具体的な教育の内容は、情報活用、情報処理、情報倫理の3本の柱からなる。情報活用ではインターネット等での情報収集の方法を扱い、情報処理ではプログラミングを扱う。これら2つについては、単なる操作方法の習得ではなく、自力で使えるようにする訓練が重要である。情報倫理では、情報社会で個人がどのように振舞うべきかの考え方を扱い、情報社会の一員としての自覚を促すものである。

電子メールの注意点

- 自分で改行する
 - 30-35文字(全角)ぐらいが適当
- 半角カタカナは使用禁止
 - 文字化けで読めない。動作不良の原因。
- 最新の機能は使わない
 - ○ 7 bit、 × 8 bit
 - HTMLメールは使わない
- Windows標準メールソフトはウィルスに注意
 - 最新版を使う。設定に注意

二進法、bit、byte

■ 二進法

$$\text{例: } (21)_{10} = \color{red}{1} \times 2^4 + \color{red}{0} \times 2^3 + \color{red}{1} \times 2^2 + \color{red}{0} \times 2^1 + \color{red}{1} \times 2^0 = (10101)_2$$
$$\color{red}{1} \times 16 + \color{red}{0} \times 8 + \color{red}{1} \times 4 + \color{red}{0} \times 2 + \color{red}{1} \times 1$$

利用: 文字については, 例えば2進数8桁を用いる場合は,

$(00000000)_2$ に文字1 を割り当てる

$(00000001)_2$ に文字2 を割り当てる

.....

$(11111111)_2$ に文字256 を割り当てる

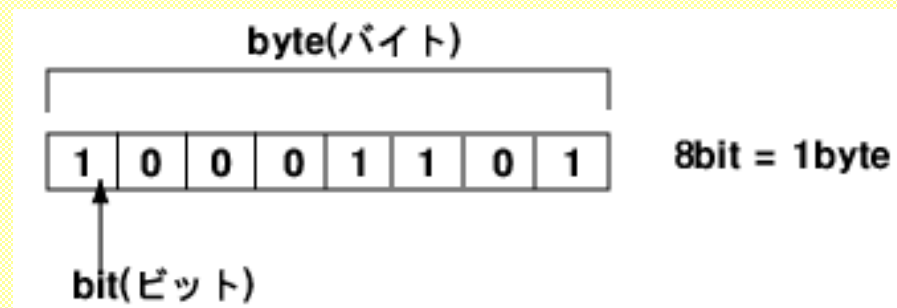
⇒256 種類の文字を割り当てることができる。

■ Bitとは

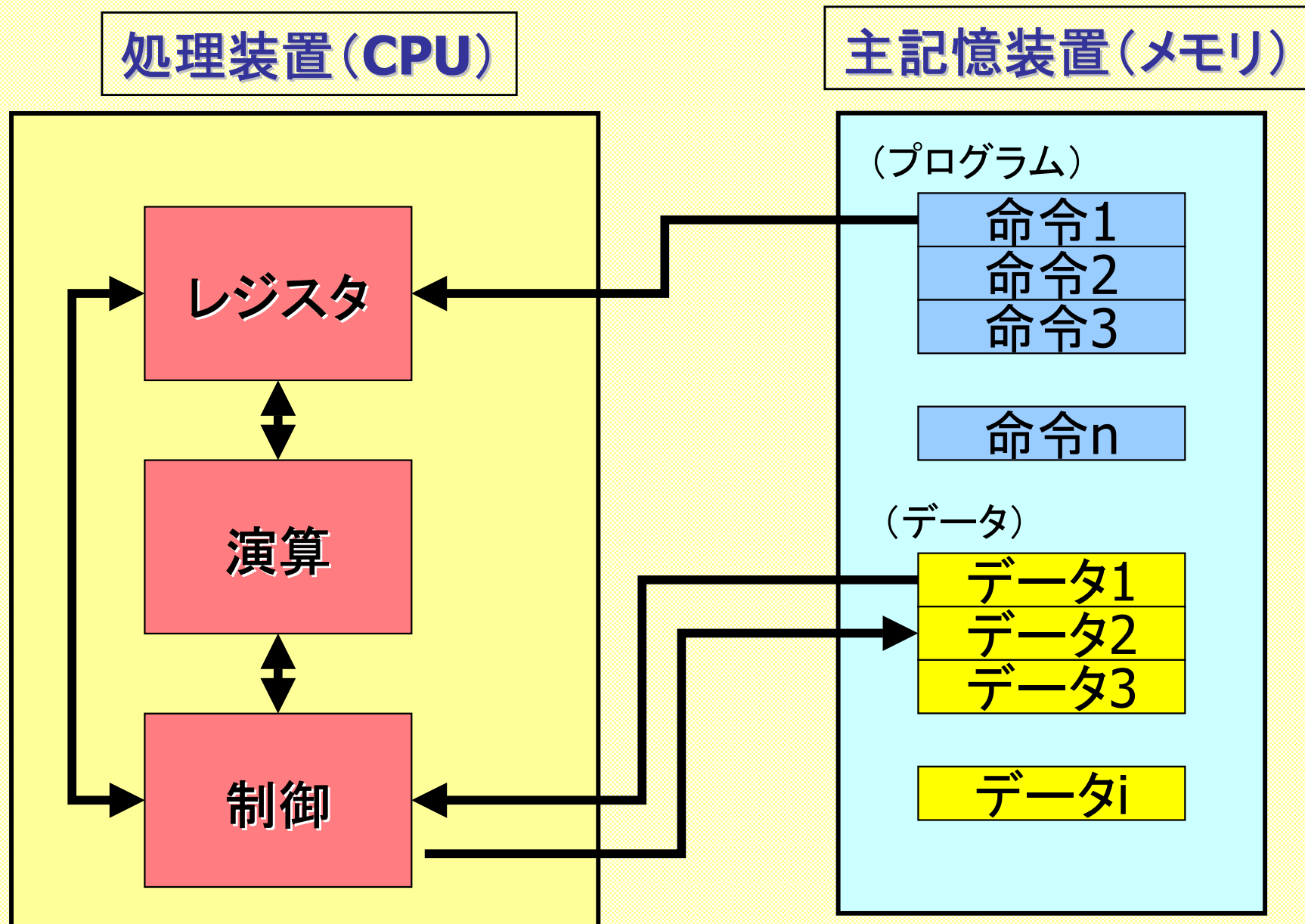
- 1桁の0または1の情報量
- ⇒2進法8桁の場合は8 bit

■ Byteとは

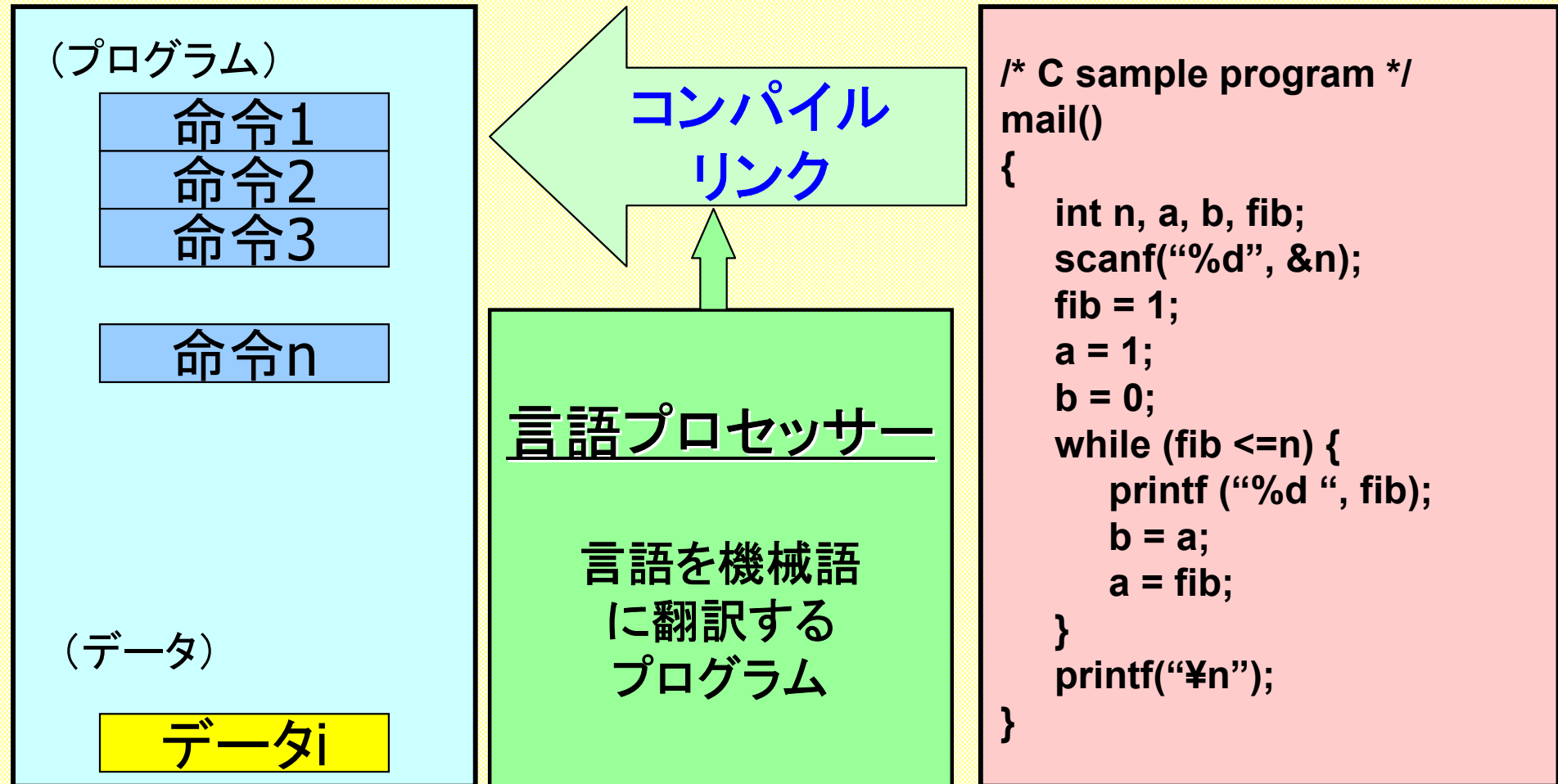
- 8 bitを8個集めたもの
- 日本語は2byteで表現
- 1024 byte = 1 k byte
- 1024 k byte = 1 M byte



計算機の仕組み(ソフトウェア)の概念図



プログラムを作るとは



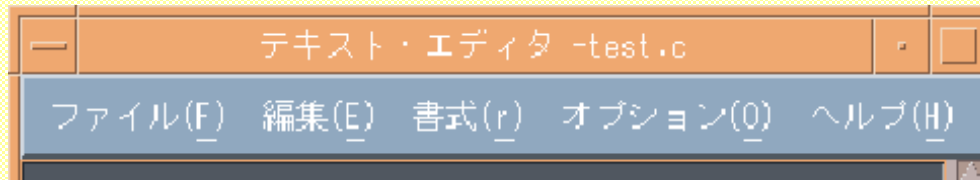
機械語: CPUがそのまま解釈
できる命令の集まり

0010 0001 011 0000 0001.....
の羅列

言語: 理解しやすくした命令の体系
C, FORTRAN, BASIC, COBOL,...

ソースプログラム:
テキスト形式で言語に基づいて
記述したプログラム

C言語でのプログラム作成の手順



```
main()
{
    printf("こんにちは¥n");
}
```

```
c2ja0rat(109)% gcc test.c
test.c: In function `main':
test.c:4: parse error before `)'
c2ja0rat(110)%
```

1. ソースプログラムの作成

テキストエディタで

- ・ソースプログラムを作成
- ・名前をつけてファイルに保存
「○×△(適当な英数字).c」
test.c

デバッグ

2. ソースプログラムのコンパイル

端末エミュレータのコマンドプロンプトで

- ・ **gcc** ○×△.c (注: テキストにはcc) と打ってReturnキーを押す。
- ・エラーメッセージが出たら
ソースプログラムを修正しやり直す。
- ・メッセージが出なかったらコンパイルは正常に終了し、a.out という実行可能プログラムが出来ている(lsで確認)。

3. プログラムの実行と動作確認

- ・ソースプログラムを作成
- ・コマンドプロンプトで a.outと打ってReturnキーを押す。

```
c2ja0rat(113)% a.out
こんにちは
```

```
c2ja0rat(110)% gcc test.c
c2ja0rat(111)% ls
a.out      intro.txt  student.txt
PC         test.c
```

コマンドの使い方(テキストp.40～) + α

- 特に重要なコマンド
 - ls(listの意味)
 - more, less, cat
 - rm(remove), cp(copy)
- 入出力切り替え(リダイレクト)
 - コマンドの出力をファイルに書き込む
a.out > test.txt
 - ファイルの内容をコマンドに送る
a.out < data.txt
- ヒストリー機能
 - history
 - !!: 直前のコマンドを実行、!n:n番目にコマンドを実行
 - CTRL + p, ↑キー: 直前のコマンドを表示。修正も可
- 時間があったら、他のコマンドやパイプも使ってみよう
- プログラム作成は、テキストエディターと端末エミュレータを同時に開いて作業を行うと便利

C言語の大まかな構造

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    int Data1, Data2;
```

```
    int Add, Sub;
```

```
    scanf("%d %d", &Data1, &Data2);
```

```
    Add = Data1 + Data2;
```

```
    Sub = Data1 - Data2;
```

```
    printf("%d %d\n", Data1, Data2);
```

```
    printf("%d %d\n", Add, Sub);
```

```
}
```

標準的な関数を利用するとの宣言

プログラムの本体であることを示す

これから始まる

整数型の変数 Data1, Data2を使う

整数型の変数 Add, Subを使う

2つのデータ(Data1,Data2)を入力

Data1とData2の和をAddに代入

Data1とData2の差をSubに代入

Data1とData2を表示して改行

AddとSubを表示して改行

終わり

ex1.c というファイル名で保存し
コンパイル、実行してみよう。

注意 ;(セミコロン)は文の区切りを示す。必須

変数と演算子の概要

■ 変数

- データをしまう箱である。
 - 名前には半角の英数字
 - mainなどのキーワードは使えない
 - 数字で始まってはいけない
 - 全角文字はダメ
- 宣言(箱を用意)してから使う。
int Data1, Data2;
- 箱には種類がある
(MS-DOSの一般的な例、UNIXはやや異なる)。

	種類	型	範囲
■	整数	int	-32768 ~ 32767
■	実数	float	$\pm 3.4 \times 10^{-38} \sim \pm 3.4 \times 10^{38}$
■	実数	double	$\pm 1.7 \times 10^{-308} \sim \pm 1.7 \times 10^{308}$
■	文字	char	文字

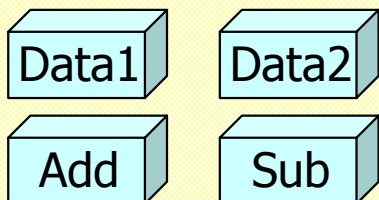
■ 算術演算子

- 足し算や引き算を行う
- +(加算), -(減算), *(乗算)
- /(除算), %(剰余)

■ 代入演算子

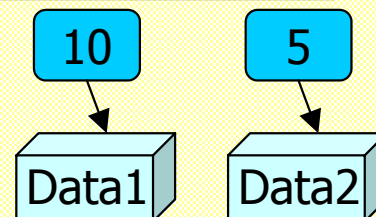
- = 代入 (数学の=とは違う!)
- += 加算代入(足してから代入)
a+=10
(aのデータに10を足した結果を新しいaのデータにする。)
a = a+10 と同じ

```
int Data1, Data2;  
int Add, Sub;
```



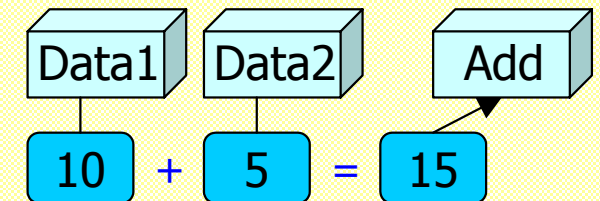
変数宣言

```
10 = Data1;  
5 = Data2;
```



代入

```
Add = Data1 + Data2;
```



算術と代入