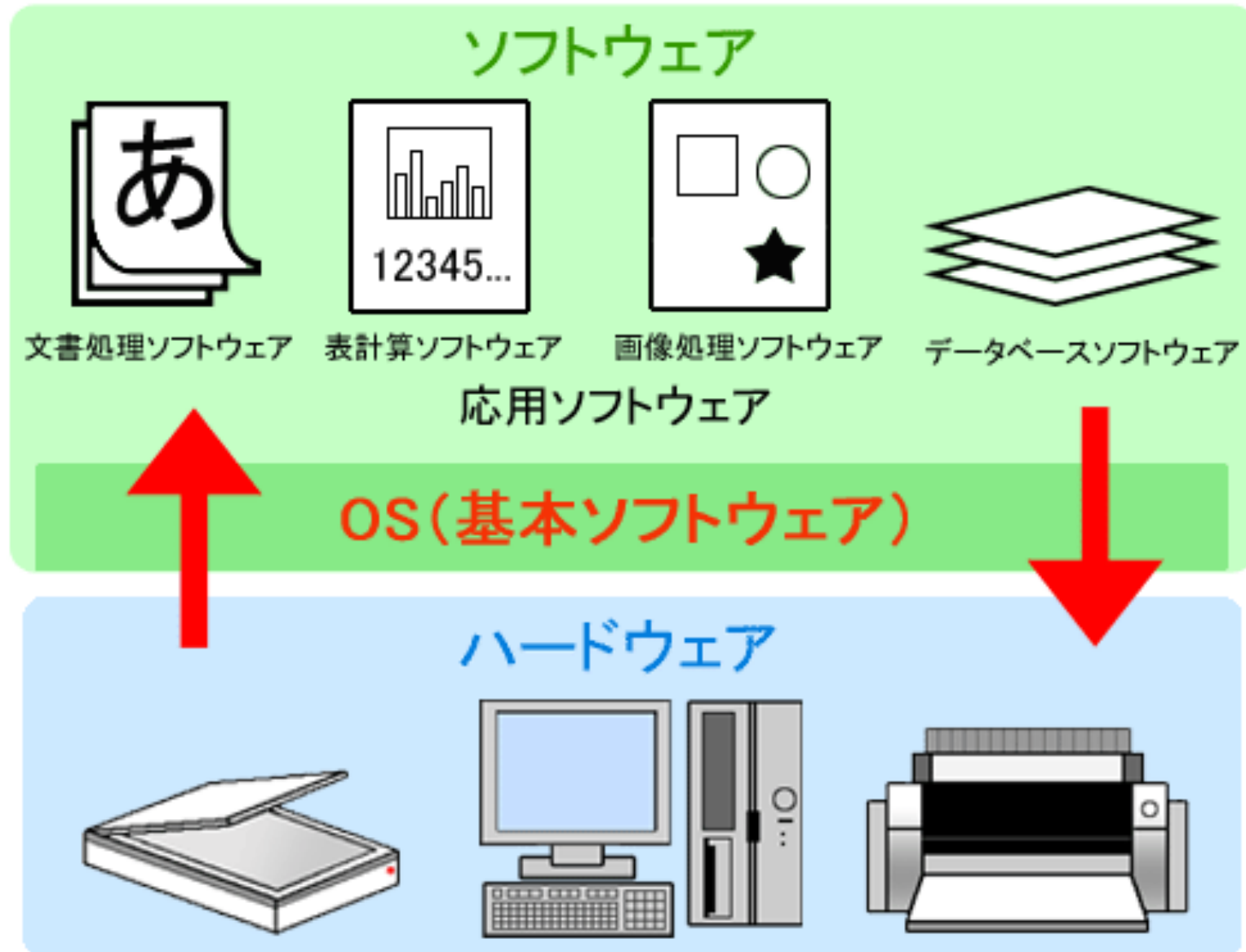


# プログラミング概論

第2週

# ハードウェアとソフトウェア



# 機械語と高級言語

- 機械語(Machine language)
  - コンピュータは0と1しか理解できない
  - 機械語はコンピュータの理解できる唯一の言語(0と1のみで記述される)
  - 人間が理解するのは困難
- C言語(Programming language C)
  - 人間が理解しやすい形で記述できる高級言語
  - コンピュータはそのままでは理解できない
- コンパイラ(Compiler)
  - 高級言語で書かれたプログラムを機械語に変換するソフトウェア

# プログラミングの流れ

プログラムを記述(Coding)



コンパイル(Compile)



実行(Execute)

# 演習室でのCプログラミング

- 端末を起動
- 以下のコマンドでテキストエディタを起動
  - `$ gedit hello.c &`
- C言語プログラムを記述
- 以下のコマンドでコンパイル  
(`cc`はコンパイルするコマンド)
  - `$ cc hello.c`
- 以下のコマンドでコンパイルされたプログラムを実行
  - `$ ./a.out`

# デバッグ

- コンパイルが一発で成功することは稀
  - よくあるバグ1: C言語の文法の沿っていない記述
  - よくあるバグ2: セミコロン(;)とコロン(:)を間違える
  - よくあるバグ3: 全角のスペースが入っている
- コンパイルエラーではまずエラーの出ている行を確認
  - コンパイラはCプログラムを上から解析されていく  
バグを見つけたらそれ以降は解析しない  
→ バグはその行もしくはその行よりも前に必ずある
  - テキストエディタの設定から行数を表示する設定を  
しておくと便利

# C言語プログラムの構成

- ▶ 以下の部分は当面「お約束」と考え、常に記述すること。  
(詳細な解説は後の項目で)

```
#include <stdio.h>

int main(void)
{
    (様々な処理)

    return(0);
}
```

- ▶ 改行や(いわゆる)半角空白は、原則として無視される。
  - 字下げなど、記述の見やすさのために利用

# printf

- ▶ 文字や数値を表示する「関数」。
  - 関数では、渡す情報をカッコの中に記述する。

```
printf    ( "Hello, World¥n" ) ;
```

- ・文字の羅列(文字列)として表現する時は""(二重引用符、ダブルクォーテーション)で囲む。
- ・¥nまたは\nは、2文字で「改行」を意味する。

- printfに限らず、命令文の最後にはセミコロン(;)を付ける。





# 演習1

- 以下のプログラムをコンパイル・実行してみよう

```
#include <stdio.h>

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

¥ではなく\と  
表示されることが  
あります。

## 演習2

- 演習1のセミコロン(;)をコロン(:)に変えとどどのようなコンパイルエラーが出るか確認してみよう
- printfをprintに変えとどどのようなエラーが出るか確認してみよう

# 次回

- 次回は演習回にします
- 演習1、演習2が終わった人はprintfを複数使ってHello, World以外の文字も表示してみよう