コンピュータ科学II

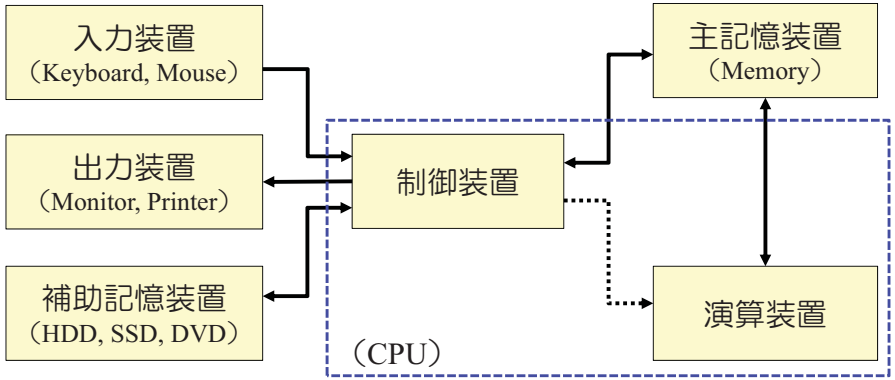
担当：武田敦志　<takeda@cs.tohoku-gakuin.ac.jp>
http://takeda.cs.tohoku-gakuin.ac.jp/

コンピュータアーキテクチャ(1)

- コンピュータの構成要素
  - 演算装置、制御装置（CPU）
    コンピュータの制御とデータの演算を行う
  - 主記憶装置（メモリ）
    計算を行うために必要なデータを記憶する装置
  - 補助記憶装置（ハードディスクなど）
    大きなデータを長期間記憶する装置
  - 入力装置（キーボードなど）
    データを入力する装置
  - 出力装置（モニタなど）
    データを出力する装置

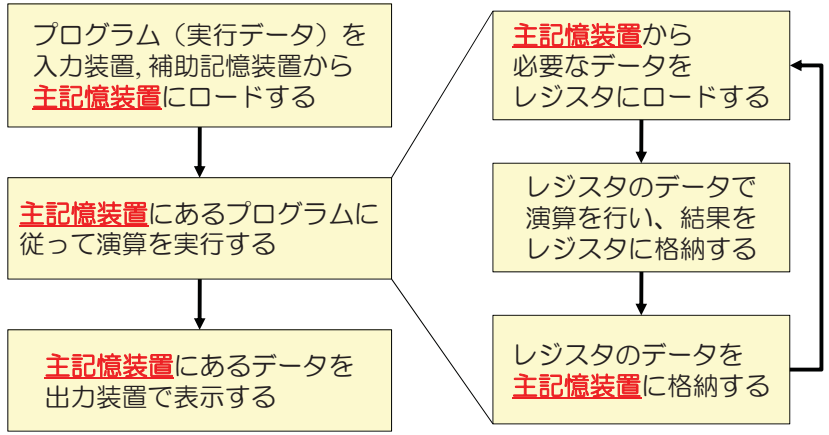
コンピュータアーキテクチャ(2)

- コンピュータの基本構成



コンピュータアーキテクチャ(3)

- コンピュータの動作の流れ



プログラム (1)

- コンピュータとプログラム

  コンピュータは**プログラムの通り**に動く

  「ストアードプログラム方式」

  ⇒ コンピュータを利用するにはプログラムが必要

  コンピュータで高度な計算を行う場合

  『どうやってプログラムを作るか？』が問題になる

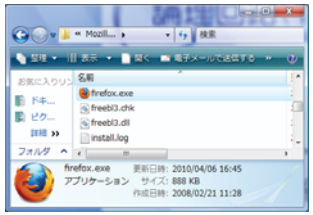
プログラム(2)

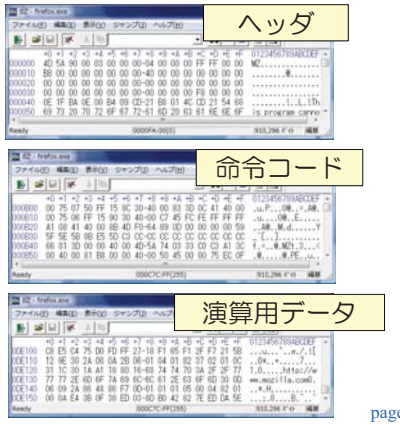
ストアードプログラム方式を発明（1946年）
John von Neumann (1903-1957)
John William Mauchly (1907-1980)
John Presper Eckert (1919-1995)

- プログラム

  コンピュータの動作を決定する**データ**

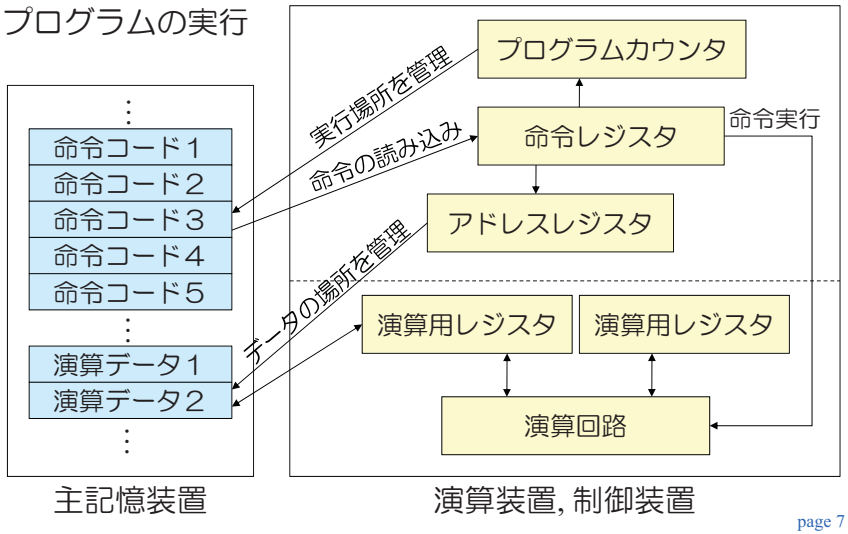
  ⇒ 命令コード（機械語）＋演算用データ





プログラム(3)

- プログラムの実行



プログラム(5)

- コンピュータウィルス

  命令コードも『データ』でしかない

  ⇒ 命令コードを書き換えることもできる

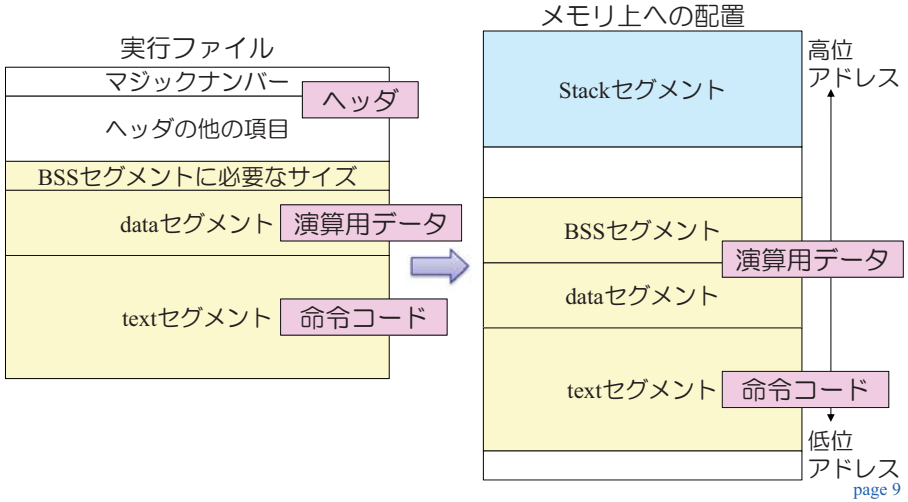
  コンピュータウィルスの多くは、
  アプリケーションの命令コードを書き換えることで
  悪意のある動作を実行する

  コンピュータウィルスはコンピュータ上で動作する

  ⇒ コンピュータウィルスも『プログラム』でしかない

実行ファイル(1)

- 実行時のアドレス空間（UNIXの場合）



実行ファイル(2)

- プログラム（実行ファイル）の作成

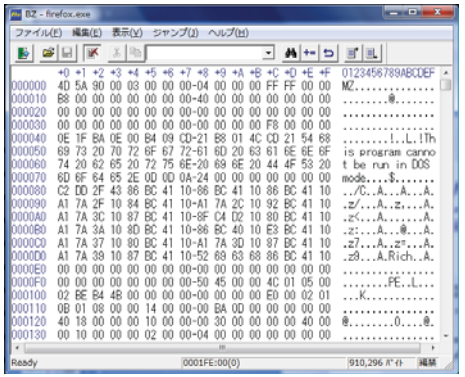
  実行ファイルを直接編集することは非常に困難

読むのが大変

⇓

書くのも大変

⇒ ソフトウェア危機



プログラミング言語(1)

- プログラミング言語

  実行ファイルの内容は『機械語＋演算用のデータ』

  コンピュータは機械語の内容に従って動作する

  人間にとって、機械語を理解することは難しい

  人が最も使いたい言語は自然言語

  人間は自然言語を使って物事を考える

  コンピュータは自然言語を理解できない

⇓

実行ファイルを作成するためにプログラミング言語が必要

プログラミング言語(2)

- プログラミング言語
  - 人が理解しやすい言語

    ⇒ 自然言語
  - コンピュータが理解しやすい言語

    ⇒ 機械語

⇓

  - 人が理解できる＋コンピュータも理解できる言語

    ⇒ プログラミング言語

通常はプログラミング言語で書かれたプログラムを機械語に翻訳して実行する

プログラミング言語(3)

- 高水準言語と低水準言語
  - 高水準言語

    人の考え方に近いプログラミング言語

    ⇒ コンピュータへの論理的な命令を表現する

    C言語，Java，Perl，Lisp，Python など
  - 低水準言語

    コンピュータの処理に適したプログラミング言語

    ⇒ コンピュータへの具体的な命令を表現する

    機械語，アセンブリ言語，中間言語

プログラミング言語(4)

- プログラミング言語の特徴
  - 人にとって読みやすい

    ⇒ 人が読み書きするための言語
  - 矛盾なく論理が記述できる

    ⇒ コンピュータで実行するための言語

プログラミング言語(5)

- 人にとって読みやすいプログラミング言語

  読みやすさの定義はプログラムの目的によって異なる

  ⇒ プログラムの目的ごとにプログラミング言語がある
  - C言語

    OSなどのシステムの開発が目的
  - Java

    高機能アプリケーションの開発が目的
  - Prolog

    人工知能プログラムの開発が目的

プログラミング言語(6)

- おまけ１（C言語のプログラム）

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include "log.h"
#include "Parameter.h"
#include "Application.h"

void print_usage(FILE *stream);
void intEvent(int sig);

static Application *App = NULL;

int main(int argc, char *argv[])
{
  Parameter *parameter;
  Application *application;
  struct sigaction saction;
  int result;

  /* set log */
  log_setLogLevel(ALL & ~DEBUG);
  log_setSyslogName(NULL);
  log_setStream(stderr);
........
```

プログラミング言語(7)

- おまけ２（Javaのプログラム）

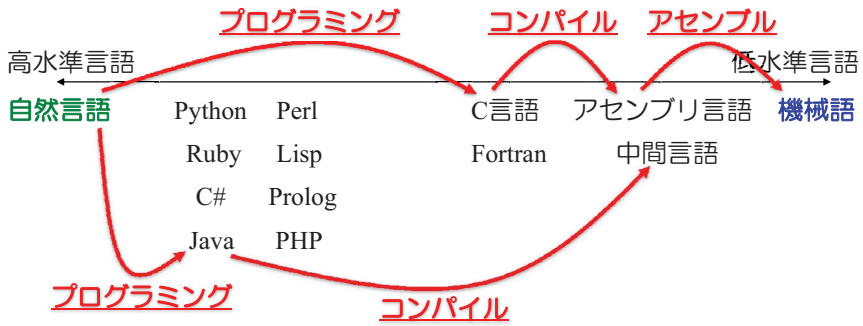
```
package jp.takedarts.util.der;
import java.util.ArrayList;
import java.util.Collection;

public class DerDecoder
{
  private DerDecoder()
  {
  }

  public static DerObject decodeObject(byte[] binary, int offset, int length)
    throws DerException
  {
    Collection<? extends DerObject> objs = decodeObjects(binary, offset, length);

    if (objs.isEmpty()) {
      throw new DerException("der binary data is not consisted");
    }
    else {
      return objs.iterator().next();
    }
  }
......
```


コンパイラ(1)

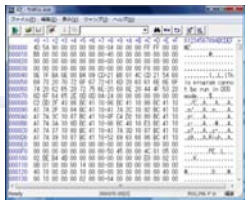
- コンパイラとアセンブラ
  - コンパイラ
    - 高水準言語を低水準言語に翻訳するソフトウェア
      - Cコンパイラ : C言語 ⇒ アセンブリ言語
      - Javaコンパイラ : Java言語 ⇒ 中間言語
  - アセンブラ
    - アセンブリ言語を機械語に翻訳するソフトウェア
      - アセンブラ：アセンブリ言語 ⇒ 機械語

コンパイラ(2)

- プログラムを作るまで



コンパイラ(3)



- 実行ファイル

プログラム（C言語）と実行ファイルの関係