

情報処理概論

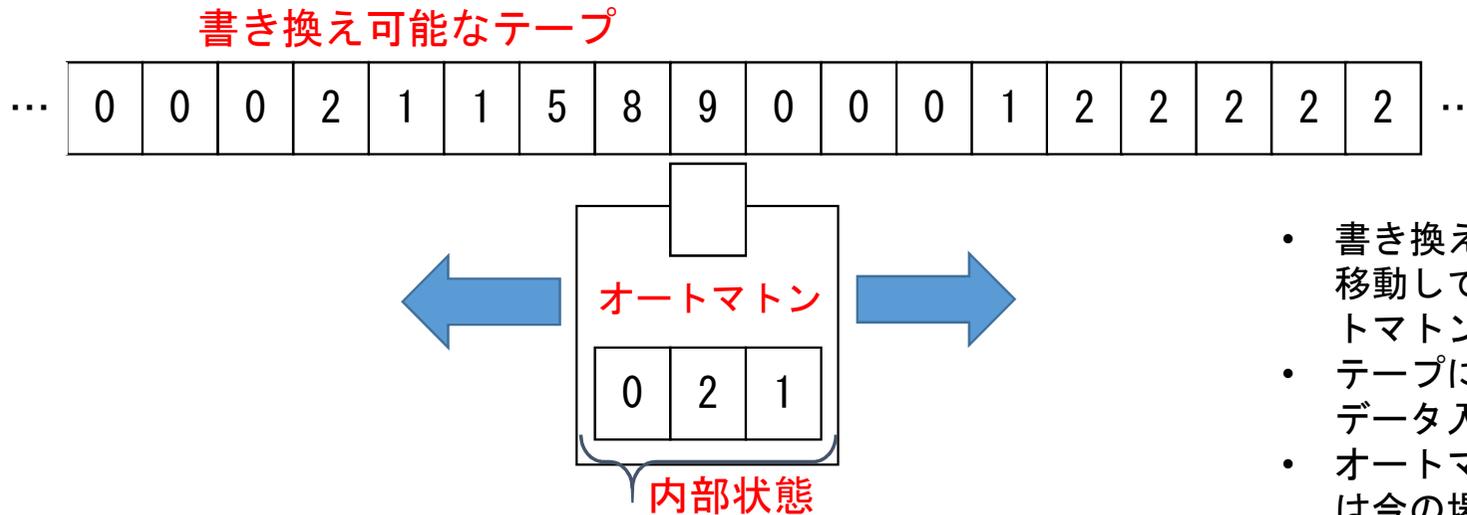
九州大学 工学部地球環境工学科 講義資料 担当:木村

01. コンピュータの原理



九州大学
KYUSHU UNIVERSITY

Turing Machine (チューリングマシン) Alan M. Turing 1936
一般的なコンピュータの数学的モデル



- 書き換え可能な無限に長いテープと、テープ上を移動してその位置のデータを読み書き可能なオートマトンから構成される。
- テープには予め文字が書かれている (計算機へのデータ入力とプログラムに相当)
- オートマトンの動作 (テープの読み書きや移動) は今の場所のデータと内部状態によって決まる。
- 処理結果はテープ上に出力される

書き換え可能なテープ = メモリー

テープ上を移動してデータを読み書きするオートマトン = CPU

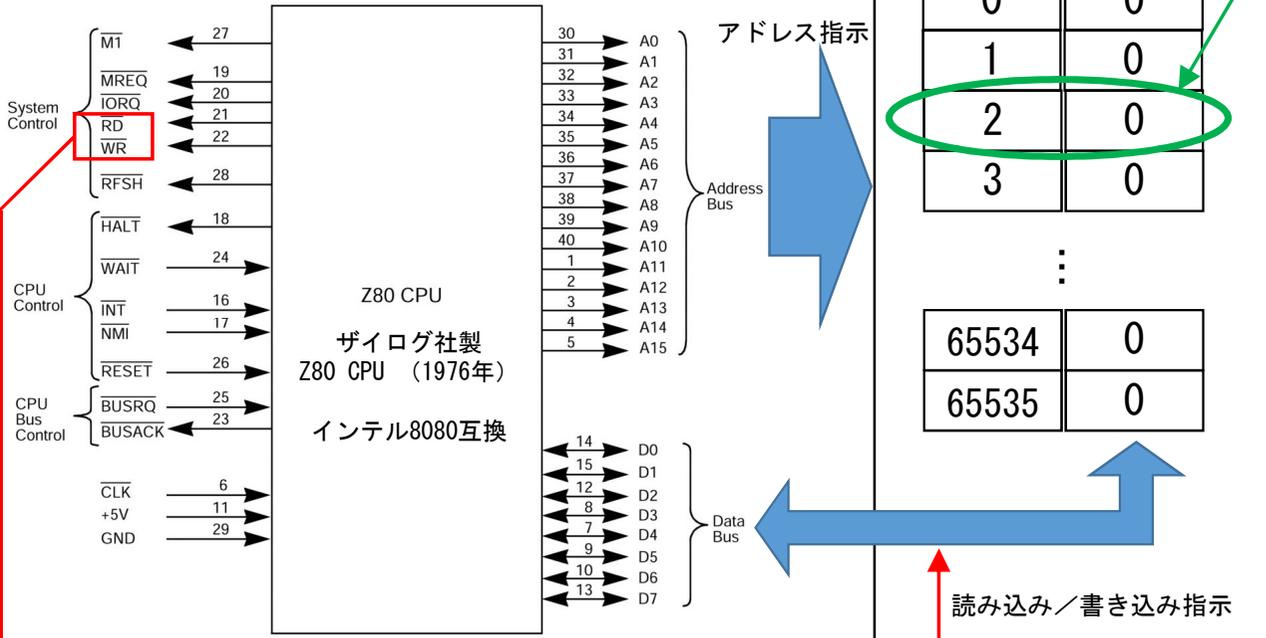
大型計算機からワンチップのマイクロプロセッサまで、あらゆるコンピュータはこれと同じ構成

CPUとメモリー

8, 16, 32 or 64 (bit)

- 1次元的に並んだメモリーがあり、メモリーに格納されたデータは**自然数の番地**で参照して読み書きできる。
- メモリーに格納されたプログラム（命令）とデータの見かけ上の区別は無い
- ノイマン型コンピュータと呼ばれる

アドレス



ある番地のデータを命令として実行したら
次の番地の命令を実行
リセット直後は0番地からスタート

Z80命令（一部）

- 0 命令を読み飛ばして次の番地の命令実行
- 58 n m CPU内のAレジスタにmn番地の値を入れる
- 50 n m mn番地にCPU内のAレジスタの値を入れる
- 128 AレジスタにBレジスタの値を加える
- 71 BレジスタにAレジスタの値を加える
- 160 AレジスタとBレジスタのANDをとり結果をAへ
- 60 Aレジスタの値を1つ増やす
- 202 n m 直前の計算結果が0のときmn番地へジャンプ
- ...

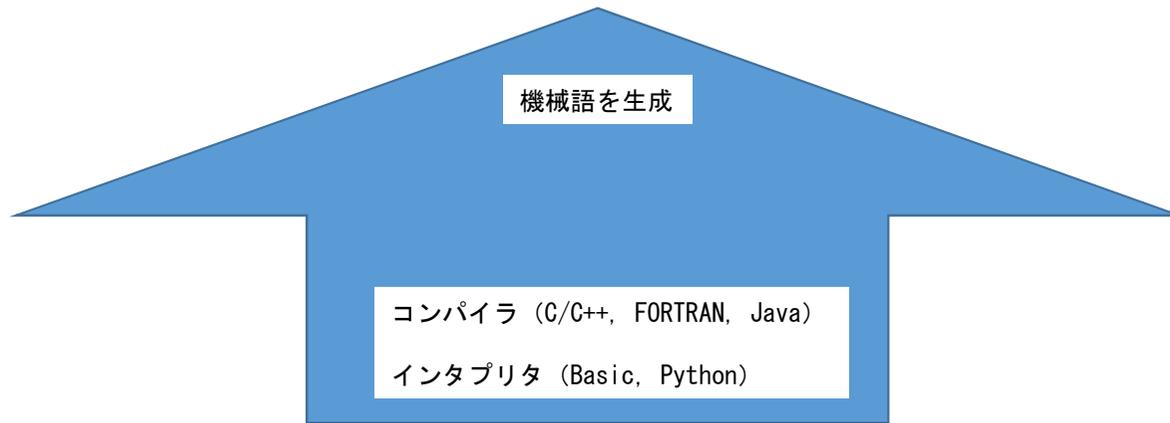
レジスタとは : CPU内の変数記憶装置
A, B, C, D, E, H, Lレジスタが使用可能だが
Aだけ特別に高機能

メモリー (64KB)

コンパイラ／インタプリタ

計算機に直接命令を指示する場合、ハードウェアに依存した命令コードやメモリの番地、使用するレジスタなどを記述しなければならない

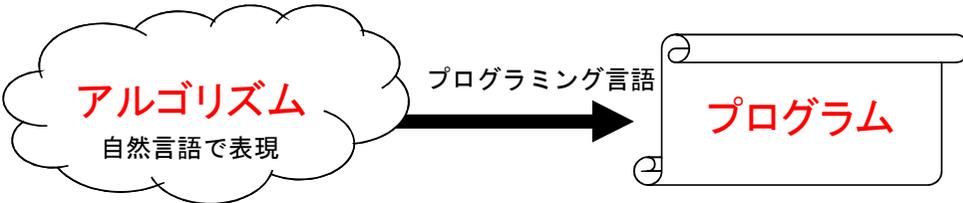
機械語＝バイナリコード(数字の羅列)／アセンブリコード(命令を簡単な文字列で表したコード)



利用者は、問題解決のための処理手順(アルゴリズム)の作成だけに専念すべき

ハードウェア非依存、かつあいまいさのないアルゴリズムの表現方法 = プログラミング言語

例) C/C++言語, FORTRAN, Basic, Java, Pythonなど



特定のハードウェアに非依存ではあるが、「アドレスを指示してデータを読み書き」するプロセッサの基本原理に沿って書式が定義されている
例) データの格納など

プログラミング言語Python

- ・ オランダ人Guido van Rossum (現Dropbox勤務) によって1980年代に開発がスタート.
- ・ インタプリタ方式 (対話式で動作可能)
- ・ Python3が最新版で主流になりつつある.
- ・ Python2もまだ現役だが2020年にサポート停止予定.
- ・ 多様な用途 (<https://www.python.org/about/apps/>)
 - ・ ウェブ, 科学技術計算, データ解析, 教育, GUI, ソフトウェア等



本講義・演習で用いるPythonの情報源

- ・ 一次情報源
 - ・ Python チュートリアル <https://docs.python.jp/3.6/tutorial/>
- ・ ウェブ上の資料・本
 - ・ Think Python
<http://greenteapress.com/wp/think-python/>
 - ・ 上記の日本語化プロジェクト
<http://www.cauldron.sakura.ne.jp/thinkpython/>
- ・ その他 (Python 関係の書籍・ウェブ上の資料は多量)
 - ・ テキスト : Pythonスタートブック 技術評論社 ISBN978-4-7741-9643-5

このテキストに沿って
実行環境を整えます

Pythonのインストール

- 1) ブラウザで次のURLへ: <https://www.python.org/>
- 2) トップページメニューから「Downloads」をクリックし、Windows→Python3の最新バージョンをクリック
ダウンロードあるいは直接実行してインストーラを起動
- 3) インストーラのウインドウの下にある「Add Python 3.6 to PATH」をクリックしてチェックを付ける
(これで環境パスの設定を自動的にやってくれる)
続いて「Install Nowをクリック」
- 4) インストールが終了したら、「Close」ボタンを押してインストーラを終了

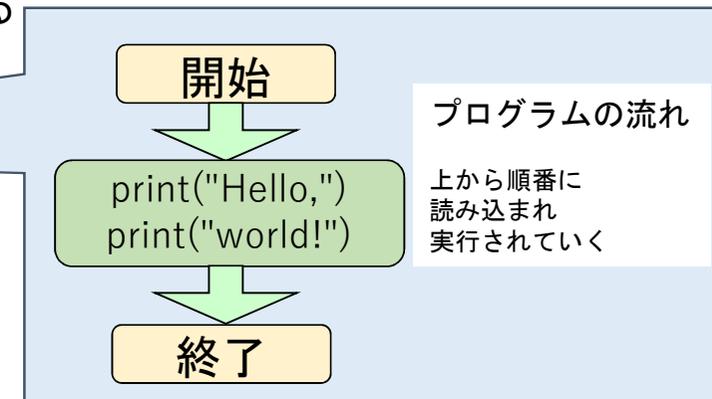
その他の設定

- 1) ファイルの拡張子が見えるようにする
スタートメニューから「Windowsシステムツール」→「コントロールパネル」
→「デスクトップのカスタマイズ」→「エクスプローラのオプション」へ進み、表示されるウインドウ
で「表示」タブをクリック
「登録されている拡張子を表示しない」の項目のチェックを外す
- 2) プログラムのコードを編集してファイル読み書きするため、
テキストエディタをインストール TeraPadがお勧め
「窓の杜」サイトからダウンロード・インストールしてください

Pythonのプログラムコードファイル編集と実行

- 1) デスクトップ、あるいはエクスプローラで「新規作成」→「フォルダ」から新しいフォルダを作成し名前を付ける。
- 2) テキストエディタを起動し、Pythonのプログラムコードを編集したら、上記で作成したフォルダへ拡張子として.py を付けて「名前を付けて保存」する。例) test.py
- 3) 「スタートメニュー」→「Windows PowerShell」をクリックして実行
または「Windowsシステムツール」→「コマンドプロンプト」を実行
- 4) 上記のシェルまたはコマンドプロンプト上で
> cd と文字入力し、半角スペースを1つ入力してから、1) で生成したフォルダをドラッグアンドドロップするとcd の文字の後ろに1) のフォルダを示すパスの文字列が入力されるので、Enterキーを押す
- 5) > dir と入力すると、カレントディレクトリのファイル一覧が表示される
- 6) > python test.py と入力してEnterキーを押すと
test.pyが実行され、結果が表示される
- 7) 以降、必要に応じてPythonのコードを編集し、再度6) でプログラムを実行

シェルのカレントディレクトリがこのフォルダへと移動します＝入出力ファイルがこのフォルダになる



画面への文字出力関数 print

- プログラム中では下のように記述

```
print("表示したい文字列")
```

引数(この内容が表示される)

- “ ” : ダブルクォート

- Python ではダブルクォートの代わりに ‘ ’ (シングルクォート) を用いてもよい

- print関数はデフォルトで改行を出力

```
print("改行するの？")  
print("改行してる！")
```

```
改行するの？  
改行してる！
```

- あえて改行させたくない時は endオプションを用いる

```
print("改行しないで", end="")  
print("改行してない！")
```

```
改行しないで改行してない！
```

- 途中で改行も出来る

```
print("途中で\n改行")
```

```
途中で  
改行
```

【注意】

日本語を使う場合は、ファイルの文字コードをUTF8にすること。

まとめ

チューリングマシン／CPUとメモリ

アルゴリズムとプログラム／コンパイラ／インタプリタ

Pythonによるプログラミング

第1回 レポート課題

print関数を4回使って「おはよう」「こんにちは」「こんばんは」「おやすみ」を各1行で表示するプログラムをprint関数を一つだけ使って実現せよ.

下記の課題提出用フォルダへ、課題の番号と提出者が分かるようにファイル名を以下のようにしてアップロードせよ
第1回1TE19xxxZ名前.py

https://share.iii.kyushu-u.ac.jp/public/IRbwAAVITI5A2X4BE45t6TqQIE0UQSQUI5Bap_kZ_sjy