

ランスタで電流の流れている状態を1、電流の流れてを0として扱い、これを組み合わせることにより、複雑うことができる。また、古典コンピュータで記憶を担では、磁性を持つ微細な物質のS極N極を 2 進数の01と存しているのだ。

図2.に示すように、この古典コンピュータの動作原理もとよりプログラマーも意識することなくコンピュータだ。これは先人たちがもたらしたコンパイラやイン技術の恩恵を享受できているためだ。コンパイル方式ミング言語（例えばC言語）では、gccなどのコンパイソースコードを古典コンピュータが理解可能な01表現ム全体をまとめて変換して実行可能な状態に変換してンタプリタ方式のプログラミング言語（例えばPython）を一つずつ機械語に翻訳しながら処理を実行してくれパイルが不要だ。図3.のようにプログラムは自動的にあるアセンブリに自動変換される。Compile(<https://godbolt.org/>)を用いると、この様子はインターを採用する言語・コンパイル方式を採用する言語などプログラミング言語で確認できる。ちなみにPythonプログランブリコードを確認したい場合はバージョン3.4以上で簡単に可能だ。Pythonが実行可能な環境でコマンド（MACならターミナル）を開き、”python”と入力し動するREPL（Read Eval Print Loop: インタラクティブ

商標、登録商標について

本書に登場する製品名などは、一般に各社の登録商標です。なお、本文中に™、®などのマークは省略しているものもあります。

フィードバック

本書の執筆については最大限の注意を払ってはおりますが、不正確な点、誤解や混乱を招くような表現、単純な誤植やなどに気づかれることがあるかもしれません。そうした場面で改善するために、フィードバックを下記までお知らせいただければ幸いです。将来の改訂に関する提案なども歓迎いたします。は以下のとおりです。

神草 経知

kamigusa.keiji@gmail.com