

システムプログラミング実験

コンパイラの作成

教員：藤田

TA：江藤 & 斎藤 & 佐藤

授業の形態

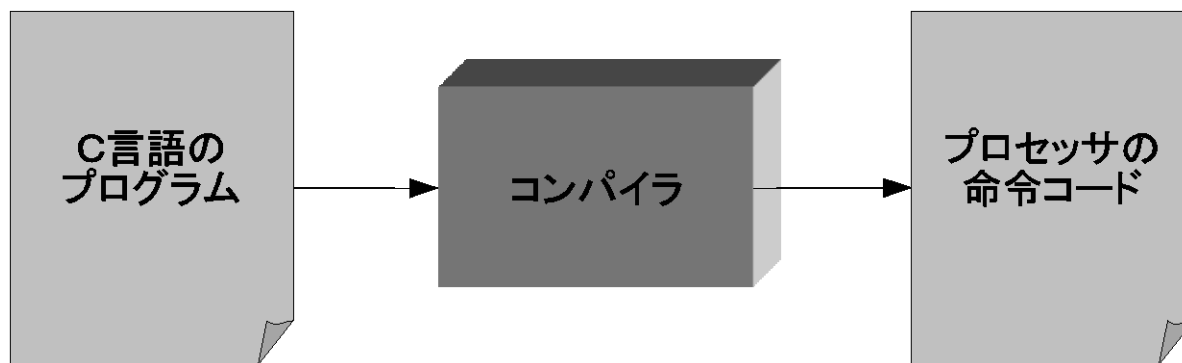
- この実験は4-405で実施しますが、課題の説明の際はZoomを利用します(4-405の音響・映像システムが良くないため)
 - Zoomの入退室記録で出欠を確認しますので、実験時間中はZoomに必ず参加してください
 - 遅刻や無断早退(Zoom途中退室)は減点対象になるので注意してください
 - ネットワークの不調などで接続に問題が発生した場合は事後に教員に報告してください(数分程度の切断なら報告は不要)
 - 教員・TAは4-405に待機していますので、質問などがある人は挙手してください
 - コロナやインフルエンザ等、正当な理由がある場合はオンライン受講を認めます(最後のスライド参照)

出席の報告

- 実験開始直後にkibacoの課題の「コンパイラ第x回出席」にアクセスし、使用中の4-405のPCの番号を報告してください。
 - オンライン受講の場合は「オンライン」と記入してください
 - 実験開始直後(30分以内)に報告すること
 - 30分以上遅れた場合はその理由も記入すること
- 遅刻・欠席する場合は事前に連絡してください。
 - e-mail: hfujita@tmu.ac.jp
 - 電車などの遅延の場合は遅延証明書, また病欠の場合は病院・薬局のレシートのコピーを(後日)提出すること

コンパイラとは？

- C言語などの高級言語のプログラムはそのままではCPUで実行できない
- CPUの命令コードやその前の中間コードに変換するプログラムがコンパイラ(gccはC言語のコンパイラ)



実験の目的

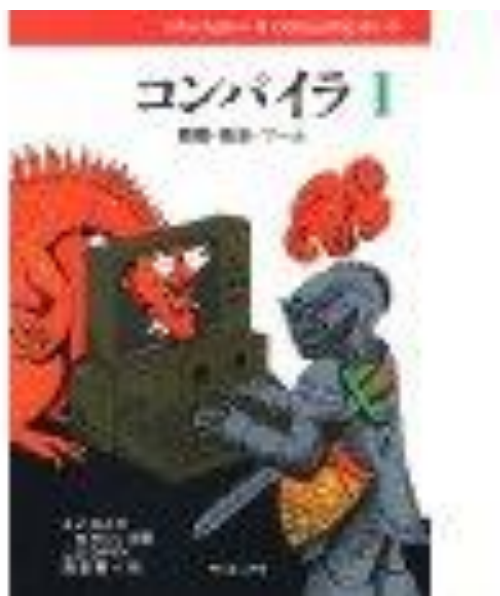
- この実験では, Pascal風の簡単な高級言語のプログラムを仮想的スタック機械の命令コードに変換するコンパイラを作成します.

関連授業

- プログラミング基礎演習Ⅱ（2年前期）
 - C言語の基本を身に付けておくこと
- データ構造とアルゴリズムⅠ・Ⅱ（1年後期・2年前期）
 - スタック, 木構造, 再帰を理解しておくこと
- 計算機システム（2年前期）
 - 計算機のハードウェアを理解しておくこと
- 形式言語とオートマトン（2年前期）
 - 正規表現, 文脈自由文法を理解しておくこと
- 言語処理系（2年後期）
 - コンパイラを深く学びたい人は履修しましょう！
- コンピュータアーキテクチャ基礎論（3年前期）

使用する教材と開発環境

- kibacoから実験資料を入手してください
 - 教科書:エイホ他著『コンパイラ I 』(通称ドラゴンブック)の第2章
- 実験の主な内容はCプログラミング
 - C言語のコンパイラとしてgccを推奨しますが, 自分の好きなCプログラミング環境を使ってもらって構いません



配付資料

- kibacoの資料のページのコンパイラ(フォルダ)の中にある「コンパイラ実験資料.zip」をダウンロードし、解凍してください。このフォルダの中には以下の資料が入っています。
 - コンパイラ実験説明スライド(このスライド)
 - コンパイラ実験指導書
 - コンパイラ I 第2章
 - 構文解析の補足
 - レポート様式
 - サンプルプログラム(フォルダ)
 - モジュール別(フォルダ)
 - lexer.c, parser.c, emitter.c, symbol.c, init.c, error.c, main.c, global.h
 - compiler.c(全モジュールを1つのファイルにまとめたもの。若干修正)

実験の内容

- この実験は全員で足並みをそろえて行うことはしません。各自で自由に進めてもらって構いません。以下は標準的な実験手順です。
 1. 『コンパイラ I』の第2章を読み、付録のプログラムを解読する。
 - 第2章の前半はコンパイラの基本的な説明が、また後半は通常の算術式を後置形に変換(例えば, 「 2×3 」 \Rightarrow 「 $23 \times$ 」)するプログラム(コンパイラ)の説明が書かれています。教科書を参考にプログラムを解読してください。
 2. 実験指導書の説明に従ってプログラムに必要なコードを追加し、目的のコンパイラを作成する。
 - 代入文とwhile文の翻訳は必ずやり遂げること
 3. プログラムが完成したら、レポートを作成する。
 - 実験指導書のレポート作成要領に従ってレポートを書いてください。作成したプログラムを自分の言葉でしっかりと説明すること

実験スケジュール

- 第1回(11月22日)
 - 課題の説明
 - 本読み
 - できればコーディング
- 第2回(11月29日)
 - コーディング
- 第3回(12月6日)
 - コーディング
 - レポートの書き方について説明
- 第4回(12月13日)
 - レポート作成

レポートの提出について

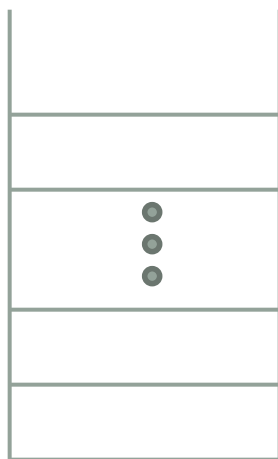
- レポート様式は参考例で、同じ内容が書かれていれば任意の様式で可 (Word/LaTeXなど好きなソフトで作成可)
- ファイル形式はPDFとすること
- レポートのファイル名は「学修番号_氏名」とすること
 - 例えば拡張子(.pdf)も含めて「12345678_日野太郎.pdf」
- 提出先: kibacoの課題のページの「コンパイラ課題」
- 提出期限: 12月20日 12:00 (遅れると減点！)

算術式を表現する代表的な方法

- 前置記法（ポーランド記法）： $+ab$
 - 演算子の次に被演算子を並べる
- 中置記法： $a+b$
 - 被演算子の間に演算子を置く
 - 数学で一般的な書き方
- 後置記法（逆ポーランド記法）： $ab+$
 - 被演算子を並べた次に演算子を置く
 - 例えば, $2 \times 3 + 4$ (中置記法) $= 23 \times 4 +$ (後置記法)
 - **スタック機械**に適した表現！

スタック機械

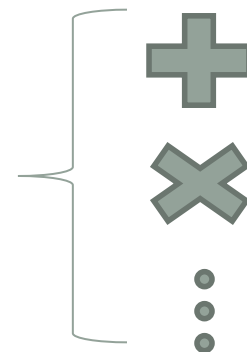
- 今回の実験では実際のCPU（インテルプロセッサなど）ではなく，仮想的スタック機械をターゲットとします．



スタック

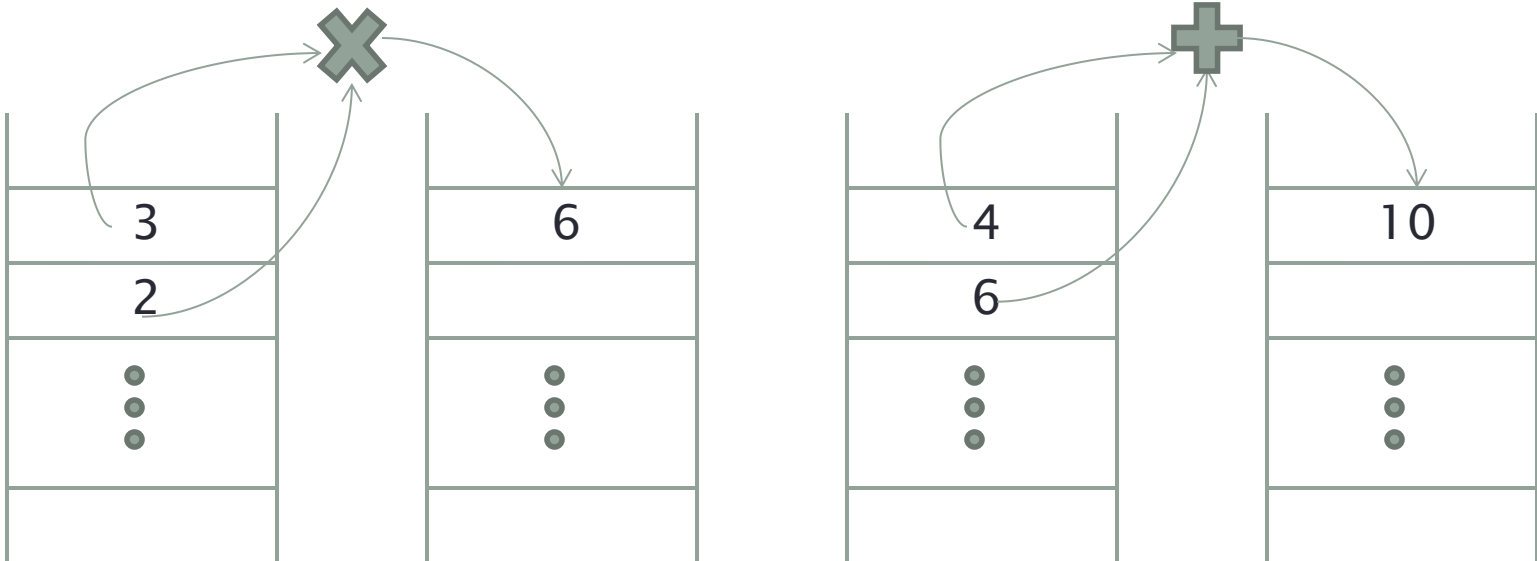


算術論理
演算装置
(ALU)



2*3+4のスタックを用いた計算

1. 2と3をスタックに積む
2. 2と3を取り出し, 積 $2*3=6$ を計算. 結果をスタックに積む
3. 4をスタックに積む
4. 4と6を取り出し, 和 $4+6=10$ を計算. 結果をスタックに積む



2*3+4のスタック機械の命令コード

1. push 2
2. push 3
3. *
4. push 4
5. +

サンプルプログラムを実行する

- このプログラムは中置形の算術式を後置形に変換します.
 1. compiler.cをコンパイルして実行ファイルを生成(例えば, UNIX環境でgccを使う場合は「gcc compiler.c」によりa.outを生成)
 - このプログラムはC言語の書き方が多少古いため, warning(警告メッセージ)が多数出力されますが, 問題なくコンパイルできます.
 2. a.outを実行すると入力待ちの状態となり, 例えば「2*3+4;」(最後のセミicolon「;」に注意!)と入力すると, その後置形が出力されます.

```
2
3
*
4
+
```

3. ファイル終了文字「EOF」を入力すると, プログラムが終了します.
 - EOFを入力するには, UNIXでは「Ctrl+D」, Windowsでは「Ctrl+Z」(どちらの場合も上手く行かないときは「Ctrl+C」)

作成するコンパイラはどんなもの？

- サンプルプログラムに必要なコードを追加して目的のコンパイラを作成します。作成したコンパイラでプログラムをコンパイルしてみましょう。
 - 例えば、代入文「`a := 2*3+4;`」を入力すると、以下の翻訳結果が出力されます。

```
lvalue a
push 2
push 3
*
push 4
+
:=
```

オンラインで受講する場合の注意点

- 口頭説明のタイミングをZoomのチャットで連絡しますので、チャット画面を立上げてメッセージを定期的に確認してください。
- 質問や困った事があればZoomのチャットで教員に連絡してください。
 - 簡単な質問はチャットで回答します。また必要があれば、ブレイクアウトルームを使って教員またはTAが個別に対応します。
 - プログラミングに関して質問する場合は、自分の開発環境(OSとCコンパイラ)の情報を付け加えてください。
 - 例えば、「Win 10でgccを使っています。コンパイルすると、〇〇のエラーが出て動きません」や「MacでClangを使っていますが、...」
- 4-405の受講者を優先しますので、オンライン受講者の質問対応が遅れることがあります。予めご了承ください。