情報工学演習 1
第 9 回

make

今日の内容

- makeについて

make

分割コンパイル（前回のおさらい）

main.c

```
#include <stdio.h>

#include “wa.h”

int main (void){
  int a, b, sum;
  a=3;
  b=4;
  sum=0;
  sum = wa( a, b );
  printf( “%d”, sum);
  return 0;
}
```

wa.c

```
int wa( int a, int b ){
    int z;
    z = a+b;
    return z;
}
```

wa. h

```
int wa( int a, int b );
```

分割コンパイル（前回のおさらい）

- main.c, wa.c, wa.h があったときのコンパイル方法
- 各.c ファイルをコンパイルする

  gcc -c main.c → main.o の生成

  gcc -c wa.c → wa.o の生成
- 統合することでmain 関数が動く

  gcc -o main main.o wa.o → main の生成

ファイルが増えると面倒！

→ make の利用

make とは？

- 複数のプログラムを一度にコンパイルするためのツール
- コンパイル，リンカ，インストールなどの作業のコマンドをテキストファイルに記述し，実行
  - make のコマンドを書いたテキストファイル：Makefile
- 利点
  - 複数のファイルを一度にコンパイル出来る
  - ユーザがプログラムの依存関係を考えてコンパイルする必要を省くことができる

make の書き方 （１つのファイル）

- １つのファイルからなるプログラムをコンパイル

hello.c

```
#include <stdio.h>
int main( void ){
   printf( “Hello world!¥n” );
   return 0;
}
```

Makefile

```
hello:
```

hello.c と同じディレクトリ

コマンドライン上
make

cc を使ってコンパイル

```
crux:sophomore yuzuko$ make
cc      hello.c   -o hello
crux:sophomore yuzuko$
```

make がしたこと

- ターゲットのhello を生成するためのソースをカレントディレクトリから探す
- hello.c を見つけるとcc を使ってコンパイル
- –oオプションを使って，実行ファイルがhello になるよう設定

コンパイラ，オプションの指定

- cc じゃなくてgcc を使いたい
  - CC=gcc
- オプションを付けてコンパイルしたい
  - CFLAGS=-Wall -O2
- リンクオプションを付けてコンパイルしたい
  - LOADLIBES = -lm

hello: の前に書く

Makefile

```
CC=gcc
CFLAGS=-Wall -O2
LOADLIBES = -lm
hello:
```

```
crux:sophomore yuzuko$ make
gcc -Wall -O2     hello.c -lm  -o hello
crux:sophomore yuzuko$
```

オプション

- コンパイラオプション（CFLAGSで指定）
  - コンパイルの内容を調節または選択可能
    - 例：–Wall 警告オプションをすべて有効にする
      - 初期化していない変数，利用していない変数などの警告
      - 他にもあるので調べてみよう
    - 例：–O2 コンパイルの最適化オプション
      - 実行ファイルのメモリや実行時間の最小化

- リンクオプション(LOADLIBESで指定）
  - 利用したいライブラリのリンクを指定
    - 例：–lm 数値演算ライブラリをリンク

make の書き方 （複数ファイル）

- 複数ファイルコンパイルの依存関係
  - main.c, wa.c, wa.h があったとき
    - main はmain.o, wa.o で作られる
    - wa.o はwa.cのみから作られる
    - main.o はmain.c のみから作られる

依存関係をMakefile に書き込めばOK!

書き方
ターゲット：ソース

複数ファイル用Makefileのサンプル

Makefile

```
CC=gcc
main: main.o wa.o
main.o: main.c
wa.o: wa.c
```

コマンドライン上で
make

```
crux:sophomore yuzuko$ make
gcc    -c -o main.o main.c
gcc    -c -o wa.o wa.c
gcc   main.o wa.o   -o main
crux:sophomore yuzuko$
```

推定機構による省略

- make は，main.o はmain.c から，wa.o はwa.c から作ると判断できる

Makefile

```
CC=gcc
main: main.o wa.o
main.o: main.c
wa.o: wa.c
```

省略前と同じようにコンパイル可能

コマンドライン上で make

```
crux:sophomore yuzuko$ make
gcc    -c -o main.o main.c
gcc    -c -o wa.o wa.c
gcc   main.o wa.o   -o main
crux:sophomore yuzuko$
```

マクロの指定

- ユーザがマクロを定義して利用可

Makefile

```
CC=gcc
main: main.o wa.o
```

Makefile

```
CC=gcc
OBJS = main.o wa.o
main: $(OBJS)
```

ターゲットの生成方法

- Makefile の書式　ターゲット：ソース
- 省略しない場合

　　ターゲット：ソース
　　コマンド

略した書き方！

コマンドは複数行にわたってOK

Makefile

```
CC=gcc
main: main.o wa.o
```

Makefile

```
CC=gcc
main: main.o wa.o
	$(CC) main.o wa.o -o main
```

.o ファイル生成と統合したプログラムの指定オプションが異なる場合
省略しないで書く必要あり

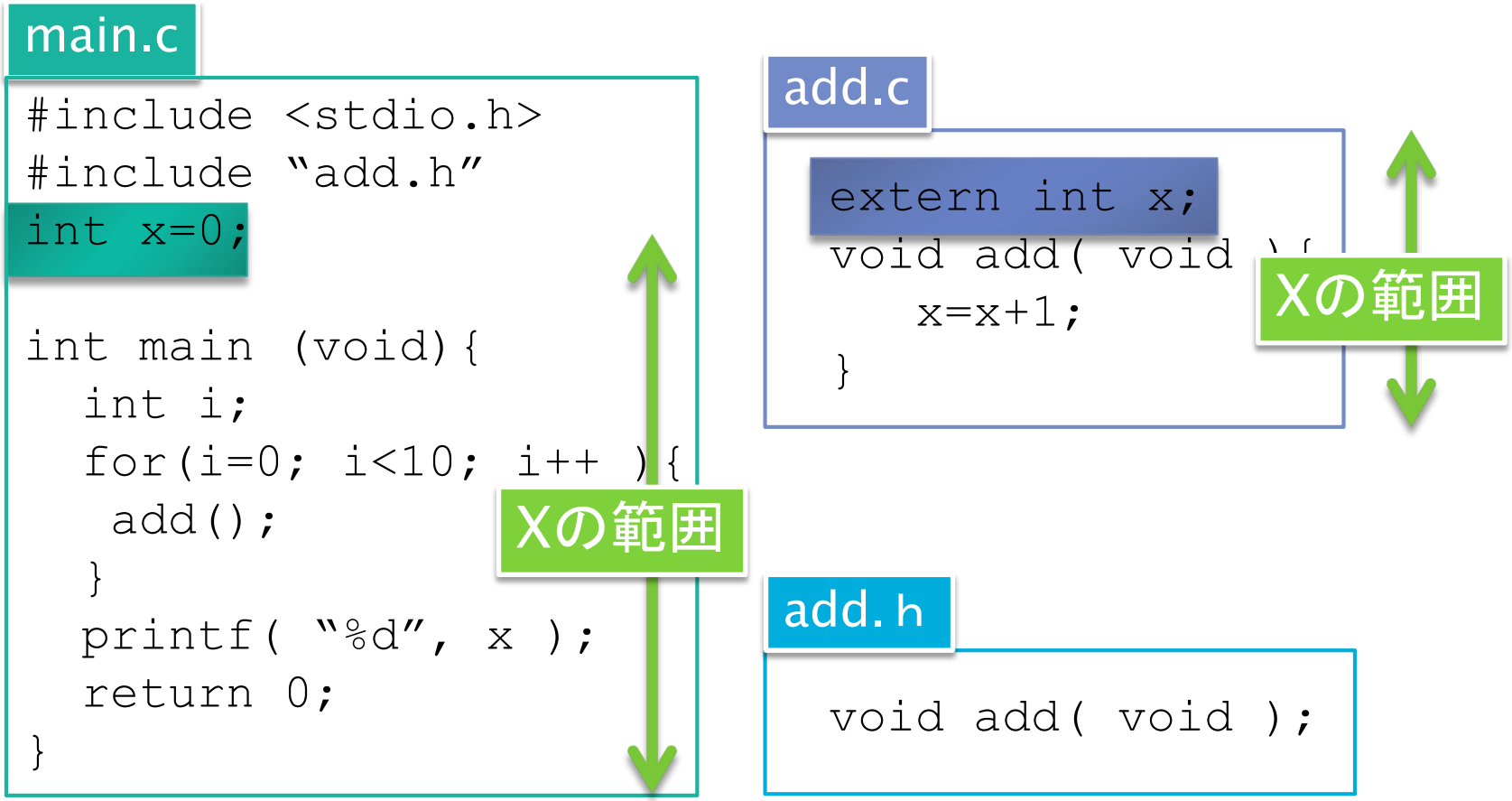
Makefile でのファイル名指定

- Makefile のファイル名がMakefile 以外の場合

  make –f Makefile のファイル名

extern 宣言

- 他のソースで定義されているグローバル変数を参照するためにする宣言のこと



第 9 回演習課題（1/2）

1. 第 8 回演習課題の 1 番目の課題のプログラムのコンパイルをmake を使って行え．そのとき用いた Makefile を提出せよ．
2. N 次元ベクトルの長さ，単位ベクトル，2つの N 次元ベクトルの内積，なす角をそれぞれ計算する関数を定義する.c ファイル，関数を宣言した.h ファイル，関数を利用する main 関数を含む.c ファイルを作成し，関数の挙動を main 関数の中で確認せよ．ただし，ベクトルの次元数 N は main 関数を含むファイルの中のグローバル変数として定義する．
   - extern を使うこと

第9回演習課題（２/２）

3. N 人の学生が N 個の研究室にそれぞれ１人ずつ配属されるとする．学生は研究室を，研究室は学生を希望順に順位付けする．このとき，この順位表をもとにして最適な配属を見いだすプログラムを作れ．

   N＝5で今回は作成して下さい．

   方針）

   ①まず学生全員が第1希望の研究室を希望し，研究室側は希望者が1名である場合は配属を内定する．希望者が重複した場合は，研究室側のリストで順位の高い学生の配属を内定する．

   ②まだ内定が出ていない学生が，以前希望したものよりも1つ順位を下げて研究室を希望する．研究室は，配属者が内定していない場合は，希望を出した学生の配属を内定する．配属者が既に内定している場合は，希望を出した学生と内定している配属者の順位を研究室側のリストで比較し，順位の高い学生の配属を内定する．

   ③全員の配属が決まるまで，②を繰り返す．

提出に関して

- 提出するもの
  - Makefile (課題1）
    - ファイル名は学籍番号 Makefile
  - ソースファイル,ヘッダファイル (.c, .h ファイル)
    - ファイル名はkadai9-学籍番号-課題番号.c

    （課題2のmain 関数があるプログラムは kadai9-学籍番号-2-main.c，関数の定義は kadai9-学籍番号-2.c，ヘッダファイルは kadai9-学籍番号-2.h）
    - ソースファイル，ヘッダファイルにはコメントアウトで学籍番号と氏名を記入

提出に関して（２／３）

- 提出するもの（続き）
  - 実行結果の出力，講義へのコメント
    - tex で作成した PDF ファイルで，レポート中に学籍番号，氏名を含む
    - ファイル名は report9–学籍番号.pdf
  - 全提出ファイルを 1 つのフォルダに格納し，そのフォルダを zip 圧縮したファイルを提出せよ.
    - フォルダ名：report9–学籍番号
    - zip圧縮したファイル名:report9–学籍番号.zip

提出に関して（３／３）

- 提出期限
  - 6 月 19 日（水）　23: 59 JST.
- 提出方法
  - Moodle から提出
- 注意点
  - ファイル名の命名規則が間違っているものは減点する
  - Repl.it の C モードで動作しないものは採点しない (0点)