

プログラミング言語基礎 [その4]

http://www.ipc.fukushima-u.ac.jp/~p058

福島県立医科大学 物理 吉田 宏
E-Mail: yoshidah@fmu.ac.jp

2004/5/10

プログラミング言語基礎(その4)

1

復習:Cygwinを起動すると

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~
\$

“~”マークは「ホームディレクトリ」

home (3)

名前	サイズ	種類
C		ファイル フォルダ
public_html		ファイル フォルダ
a.exe	19 KB	アプリケーション
ex0001.c	1 KB	C ファイル

2004/5/10

復習:ディレクトリの移動

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~
\$ cd C
yoshidah@PROCYON ~/C
\$

ディレクトリ「C」に移動させる
「C」に移動したことを示している

C

名前	サイズ	種類
Lec01		ファイル

2004/5/10

復習:もう1つ下のディレクトリへ

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~
\$ cd C
yoshidah@PROCYON ~/C
\$ cd Lec01
yoshidah@PROCYON ~/C/Lec01
\$

“/”はディレクトリの区切り

~/C:
ホームディレクトリ (~/) の下の “c”

~/C/Lec01:
“~/c” の下の “Lec01”

Lec01

名前	サイズ	種類
ex0101.c	1 KB	C ファイル
ex0102.c	1 KB	C ファイル
ex0103.c	1 KB	C ファイル
ex0104.c	1 KB	C ファイル
ex0105.c	1 KB	C ファイル

2004/5/10

復習:1つ上のディレクトリへ

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~/C/Lec01
\$ cd ../
yoshidah@PROCYON ~/C
\$

“../”は1つ上のディレクトリ

XXXXXXX@IPC12 ~/C/Lec01
\$ cd ../
XXXXXXX@IPC12 ~/C
\$

C

名前	サイズ	種類
Lec01		ファイル

2004/5/10

復習:一気にホームディレクトリへ

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~/C/Lec01
\$ cd
yoshidah@PROCYON ~
\$

XXXXXXX@IPC12 ~/C/Lec01
\$ cd
XXXXXXX@IPC12 ~
\$

home (3)

名前	サイズ	種類
C		ファイル フォルダ
public_html		ファイル フォルダ
a.exe	19 KB	アプリケーション
ex0001.c	1 KB	C ファイル

2004/5/10

復習:一気に目的のディレクトリへ

Now bash version 2.05b.0(9)-release start!
yoshidah@PROCYON ~
\$ cd C/Lec01
yoshidah@PROCYON ~/C/Lec01
\$

XXXXXXX@IPC12 ~
\$ cd C/Lec01
XXXXXXX@IPC12 ~/C/Lec01
\$

Lec01

名前	サイズ	種類
ex0101.c	1 KB	C ファイル
ex0102.c	1 KB	C ファイル
ex0103.c	1 KB	C ファイル
ex0104.c	1 KB	C ファイル
ex0105.c	1 KB	C ファイル

2004/5/10

復習:ファイルのコピー

XXXXXXX@IPC12 ~
\$ cp ex0001.c ex0101.c
XXXXXXX@IPC12 ~
\$ ls
a.exe ex0001.c ex0001.exe
ex0101.c public_html
XXXXXXX@IPC12 ~
\$

2004/5/10

復習:別のディレクトリへ ファイルをコピー

```

XXXXXXXX@IPC12 ~ ファイルを~/から~/C/Lec01にコピー
$ cp ex0101.c C/Lec01/ex0101.c
XXXXXXXX@IPC12 ~
$ cd C/Lec01 ← /C/Lec01に移動
XXXXXXXX@IPC12 ~/C/Lec01
$ ls ← ファイルを確認
ex0101.c
XXXXXXXX@IPC12 ~/C/Lec01
$
    
```

2004/5/10

プログラミング言語基礎(その4)

9

復習:ファイルの移動

```

XXXXXXXX@IPC12 ~
$ mv ex0101.c C/Lec01/ex0101.c
XXXXXXXX@IPC12 ~ ファイルを~/から~/C/Lec01に移動
$ ls ← ファイルを確認
a.exe ex0001.c ex0001.exe
public_html
XXXXXXXX@IPC12 ~
$ cd C/Lec01 ← /C/Lec01に移動
XXXXXXXX@IPC12 ~/C/Lec01
$ ls ← ファイルを確認
ex0101.c
    
```

2004/5/10

プログラミング言語基礎(その4)

10

復習:ファイルの新規作成

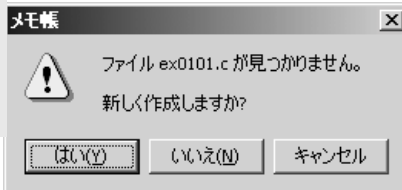
```

XXXXXXXX@IPC12 ~/C/Lec01
$ notepad ex0101.c &
XXXXXXXX@IPC12 ~/C/Lec01
$
    
```

notepad のあとに必ずファイル名をつける

~/C/Lec01 に新たに ex0101.c を作成するとき

「はい(Y)」をクリック



2004/5/10

プログラミング言語基礎(その4)

11

復習:ファイルの編集

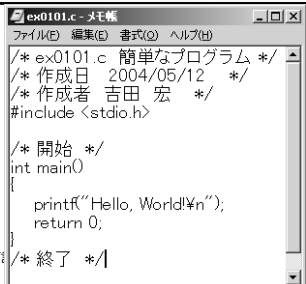
```

XXXXXXXX@IPC12 ~/C/Lec01
$ notepad ex0101.c &
XXXXXXXX@IPC12 ~/C/Lec01
$
    
```

notepad のあとに必ずファイル名をつける

既に ~/C/Lec01 にある ex0101.c を編集するとき

必要な部分の編集をする



2004/5/10

プログラミング言語基礎(その4)

12

復習:プログラムのコンパイル

```

XXXXXXXX@IPC12 ~/C/Lec01
$ gcc ex0101.c
XXXXXXXX@IPC12 ~/C/Lec01
$ ls ← ファイルを確認
a.exe ex0101.c
XXXXXXXX@IPC12 ~/C/Lec01 賢いコンパイル
$ gcc -o ex0101 ex0101.c
XXXXXXXX@IPC12 ~/C/Lec01
$ ls ← ファイルを確認
a.exe ex0101.c ex0101.exe
    
```

2004/5/10

プログラミング言語基礎(その4)

13

効率よく ex0102.c を作る (ex0101.c から ex0102.c を作る)

```

XXXXXXXX@IPC12 ~/C/Lec01
$ cp ex0101.c ex0102.c
XXXXXXXX@IPC12 ~/C/Lec01
$ ls ← ファイルを確認
ex0101.c ex0101.exe ex0102.c
XXXXXXXX@IPC12 ~/C/Lec01
$ notepad ex0102.c &
XXXXXXXX@IPC12 ~/C/Lec01
$
    
```

notepad (メモ帳) を起動して必要な部分の編集をする

今日の目標!

- 出力関数 "printf" をマスターする
 - printf 関数の使い方
- 入力関数 "scanf" を使ってみよう
 - scanf 関数の使い方

2004/5/10

プログラミング言語基礎(その4)

15

printf 関数

```
printf("Hello, World!\n");
```

"" で囲まれた書式で出力する
"Hello, World!\n" 書式を指定する

Try 課題 1.6~1.8 !!

```

課題 1.6 printf("Hello, World!\n");
→ printf("Hello, ");
printf("World!\n");
課題 1.7 printf("Hello, World!\n");
→ printf("Hello, \nWorld!\n");
課題 1.8 printf("Hello, World!\n");
→ printf("Hello, \n\nWorld!\n");
    
```

問題: %n は何を表す記号?

2004/5/10

プログラミング言語基礎(その4)

16

出力の書式を指定する

```
int y=16,m=4,d=21;
printf("平成%2d年%2d月%2d日¥n",y,m,d);
```

表示する桁数を指定

y,m,dの値を"....."のように出力せよ

「参考1 書式付出力関数 printf」(p.8)を読んで

Try 課題 1.9 !!

2004/5/10

プログラミング言語基礎(その4)

17

printf 関数の書式指定

変数型	宣言の仕方	printfでの使い方
整数	int i;	printf("%d¥n",i);
文字	char c;	printf("%c¥n",c);
文字列	char a[10];	printf("%s¥n",a);
単精度実数	float x;	printf("%f¥n",x); printf("%e¥n",x);
倍精度実数	double y;	printf("%f¥n",y); printf("%e¥n",y);

2004/5/10

プログラミング言語基礎(その4)

18

何か計算をするプログラムを作ろう

ある指定した x の値に対して数式 $y=f(x)$ を計算するプログラムを作ろう!!

$$x \mapsto y = x^2 + 2x + 1$$

$$x=1 \mapsto y_1 = 1^2 + 2 \times 1 + 1 = 4$$

$$x=2 \mapsto y_2 = 2^2 + 2 \times 2 + 1 = 9$$

注意 プログラムの中では

- 積を * で表す
- x の2乗は x * x で表す
- = は「右の値を左の変数に代入」を意味する

2004/5/10

プログラミング言語基礎(その4)

19

```
1:/* ex0201.c y=x*x+2*x+1を計算するプログラム */
2:/* 作成日 2004/4/28 */
3:/* 作成者 xxxxxx */
4:#include <stdio.h>
5:
6:/* 開始 */
7:int main()
8:{
9:    int x,y; /*x,yを整数型変数として宣言。(宣言文)*/
10:
11:    printf("xを入力してください");
12:    scanf("%d",&x); /* xに値を整数型のデータとして入力*/
13:
14:    y=x*x+2*x+1; /* yに値を代入 (代入文) */
15:
16:    printf("xが%dのとき, y=x*x+2*x+1は %d です¥n",x,y);
17:    return 0;
18:}
19:/* 終了 */
```

include 文 とりあえずは「おまじない」

2004/5/10

プログラミング言語基礎(その4)

20

scanf 関数

```
int x;
...
scanf("%d",&x);
```

""で囲まれた書式で入力する
"%d"の書式で変数xに入力値を代入する。
代入する変数 "x" の前に必ず "&" をつける

```
int yy,mm,dd;
scanf("%d-%d-%d",&yy,&mm,&dd);
```

入力データ 2004-5-12

2004/5/10

プログラミング言語基礎(その4)

21

3. 変数(データ型)

型	データ	バイト幅	範囲
char	文字	1byte(8bit)	文字 or -128~127の整数
int	(符号付)整数	4byte(32bit)	$-2^{31} \sim 2^{31} - 1$ の整数
float	単精度実数	4byte(32bit)	約 $10^{-38} \sim 10^{38}$ (7桁有効)
double	倍精度実数	8byte(64bit)	約 $10^{-308} \sim 10^{308}$ (15桁有効)

2004/5/10

プログラミング言語基礎(その4)

22

printfとscanf

変数型	宣言文	printfの中身	scanfの中身
整数	int i;	printf("%d¥n",i);	scanf("%d",&i);
文字	char c;	printf("%c¥n",c);	scanf("%c",&c);
文字列	char a[10];	printf("%s¥n",a);	scanf("%s",&a);
単精度実数	float x;	printf("%f¥n",x);	scanf("%f",&x);
倍精度実数	double y;	printf("%f¥n",y);	scanf("%lf",&y);

2004/5/10

プログラミング言語基礎(その4)

23