

## 第2回 計算機基礎実習II

座席指定はありません。

Linux を起動して下さい。

setup コマンドを（再度）実行して下さい。

```
t180900@s01cd0542-160:~$ setup
```

計算機基礎実習II 2018 のウェブページから、  
以下の課題に自力で取り組んで下さい。

第1回の復習課題(rev01)

第2回の基本課題(base02)

## 画面共有コマンド tssc

準備：再度、setup コマンドを実行して下さい。

```
t180900@s01cd0542-160:~$ setup
```

tssc コマンドを実行

通常ウィンドウで起動：t180900@s01cd0542-160:~\$ tssc

全画面で起動：t180900@s01cd0542-160:~\$ tssc -f

半画面で起動：t180900@s01cd0542-160:~\$ tssc -h

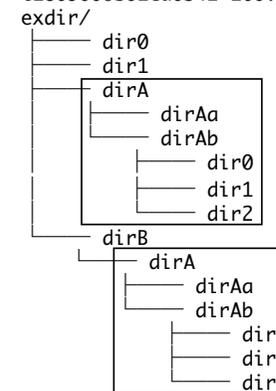
tssc コマンドを停止：

Ctrl-C（コントロールキー + C）で tssc を止める

## 第1回課題の回答例

### ex01-1.sh

```
t180900@s01cd0542-160:~/kiso2-2018$ tree exdir
```



15 directories, 0 files

ex01-1.sh の回答例：

```
mkdir kiso2-2018/exdir
cd kiso2-2018/exdir
mkdir dir0 dir1 dirA dirB
cd dirA
mkdir dirAa dirAb
cd dirAb
mkdir dir0 dir1 dir2
cd ~/kiso2-2018/exdir
cp -r dirA dirB/.
```

## ex01-2.txt

```
t180900@s01cd0542-160:~$ cp -r /etc/texmf ~/kiso2-2018/texmf
t180900@s01cd0542-160:~$ cd kiso2-2018
t180900@s01cd0542-160:~/kiso2-2018 $ tree ~/kiso2-2018/texmf
/home/t180900/kiso2-2018/texmf
├── dviptdms
├── dvips
├── config
├── fmt.d
│   ├── 00tex.cnf
│   ├── 10texlive-base.cnf
│   ├── 10texlive-lang-cjk.cnf
│   └── 10texlive-lang-cyrillic.cnf
├── :
├── texmf.d
│   ├── 00debian.cnf
│   └── 96JadeTeX.cnf
├── web2c
│   ├── mktex.cnf
│   └── texmf.cnf
└──
```

11 directories, 31 files

## ex01-3.txt

ホームディレクトリの下に kiso2-2018 ディレクトリが存在し、  
(mkdir kiso2-2018)  
その kiso2-2018 ディレクトリには2つのディレクトリ ex00 と ex01 のみが  
存在している。  
cd kiso2-2018  
mkdir ex00 ex01  
ex00 ディレクトリの下には、foo という名前のテキストファイルが1つだけ存在し、  
foo の内容(中身)は /etc/lsb-release というファイルと同じである。  
cp /etc/lsb-release ex01/foo  
ex01 ディレクトリの下には、ファイルやディレクトリは存在しない。

```
t180900@s01cd0542-160:~/kiso2-2018$ cat ex00/foo
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.5 LTS"
t180900@s01cd0542-160:~/kiso2-2018$ tree
.
├── ex00
│   └── foo
└── ex01

2 directories, 1 file
t180900@s01cd0542-160:~/kiso2-2018$
```

## ex01-4.sh

ex01-4.sh の回答例 :

```
mkdir ~/kiso2-2018/$1
echo ~/kiso2-2018/$1 を作成しました。
```

第1回

今回 (第2回)

“Linuxの使い方に慣れ、C言語による簡単なプロ  
グラミングができるようになる。”

-計算機基礎実習II 2018 シラバス「到達目標」

## いくつかのお知らせ

Visual Studio Code の使い方まとめ

(GUIからVisua Studio Code を起動する)

オンライン・プログラミング環境 (paiza.io)

「理工出席」アプリのテスト運用

## 第2回の基本課題

## ex02-1.c から実行ファイルを作る

```
#include <stdio.h>
int main() {
    int a;
    printf("整数を入力して下さい: ");
    scanf("%d", &a);

    printf("入力された値は %d です.\n", a);
    return 0;
}
```

C言語のソースファイル  
||  
単なるテキストファイル

### コンパイル

### リンク

ex02-1.c の  
オブジェクトファイル

### ex02-1.c の 実行ファイル

printf() や scanf() の  
オブジェクトファイル

## Cコンパイラ : cc (gcc)

C言語ソースファイルをコンパイル・リンクして  
実行ファイルを作成するコマンド /usr/bin/cc

ソースファイル名      実行(オブジェクト)ファイル名

```
t180900@s01cd0542-160:~$ cc ex02-1.c -o ex02-1
t180900@s01cd0542-160:~$ ./ex02-1
```

「このディレクトリの」 ex02-1を実行

実行ファイル名を省略すると a.out が作られる

```
t180900@s01cd0542-160:~$ cc ex02-1.c
t180900@s01cd0542-160:~$ ./a.out
```

「このディレクトリの」 a.out を実行

## ex02-1.c

Cの標準関数 printf() と scanf() の定義を読み込む

```
#include <stdio.h>
```

```
int main() { Cのプログラムは、必ず main() 関数から開始される  
            整数値の変数 a を定義  
    int a;   画面に「整数を入力して下さい: 」と出力  
    printf("整数を入力して下さい: ");  
            キーボードからの入力を整数値として変数 a に代入  
    scanf("%d", &a);  
            画面に「入力された値は〇です。 \n」と出力  
    printf("入力された値は %d です。 \n", a);  
  
    return 0; main()関数は 0 を返す (次回の話題)  
}
```

main() 関数の終わり

## 入力と出力

(キーボードから)

“3” を入力

**“3”**

scanf()

ex02-1.c の  
実行ファイル

printf()

**“入力された値は 3 です。”**

入力にもとづいた (画面) 出力

## printf() : 文字列を画面に出力する関数

詳細は後日

前期の tPrintf() 関数と名前が違うだけ。

printf() の括弧内の文字列を画面に出力する。

関数の引数 (カッコ内) の文字列を出力する

```
printf("整数を入力して下さい: ");
```

文字列なのでダブルクォーテーション "が必要

scanf() の第1引数の書式にしたがって、第2引数以降の変数の値を出力する。

文字列に変換指定子(この場合は整数指定)があると、

```
printf("入力された値は %d です。 \n", a);
```

第2引数以降の値を指定された型だと仮定してその位置に表示する

## scanf() : キーボードから数値などを入力する関数

詳細は後日

前期の tScanf() 関数と名前が違うだけ。

scanf() の第1引数の書式にしたがって、第2引数以降の変数に値を代入する。

入力されるものが変換指定子に従うと仮定して、

```
int a;   この場合は、1つの整数を表すような文字列  
scanf("%d", &a); 変数名前に&を付ける(理由はいずれ)
```

その値を変数に代入する

文字列なのでダブルクォーテーション "が必要