第01回の演習課題

基本課題 ex01-0.txt

以下のリンクにあるアンケートフォームに回答して下さい。

C言語についての理解度調査その1 kiso2-2019

「C言語に関する単語や概念について、現在の理解度を「自分が」どのように感じているかお答え下さい。」

回答が終わったらフォーム画面下の「送信(Submit)」ボタンを忘れずに押して下さい。

アンケートに回答後、Visual Studio Code などのテキストエディタを用いて、

1. *学籍番号*
2. *日本語氏名*

を含む適当な自己紹介文をテキストファイル ex01-0.txt として作成し、kiso2 コマンドを用いて提出しなさい。

演習課題 ex01-1.sh

次の**tree コマンド**の実行例は、**カレントディレクトリが ~/kiso2-2019 であるときのターミナル上での出力例である。**

```
t190900@s01cd0542-160:~/kiso2-2019$ tree exdir
exdir/
├── dirA
│   ├── dirAa
│   └── dirAb
│       ├── dir0
│       ├── dir1
│       └── dir2
├── dirB
│   └── dirA
│       ├── dirAa
│       └── dirAb
│           ├── dir0
│           ├── dir1
│           └── dir2
├── dirX
└── dirY

15 directories, 0 files
```

このようなディレクトリ構造を作成するために必要となるシェルコマンドを、テキストファイル ex01-1.sh に入力・保存し、kiso2 コマンドを用いて提出しなさい。 ただし、最初のカレントディレクトリがホームディレクトリであることを仮定してよい。

また、ex01-1.shに入力するシェルコマンドはコマンドごとに改行し、1行に1つのコマンドのみが入力されるものとする。

たとえば、ホームディレクトリから

1. **ディレクトリ exdir を作成し、**
2. **続いて exdir に移動、**
3. **さらにディレクトリ dirX を作成**

する場合、ex01-1.sh は次のような内容となる。

```
mkdir kiso2-2019/exdir
cd kiso2-2019/exdir
mkdir dirX
 :
```

演習課題 ex01-2.txt

/etc/texmf ディレクトリを、~/kiso2-2019/texmf (*ホームディレクトリの下の、kiso2-2019 ディレクトリの下の、texmfディレクトリ*)に、コピー元のディレクトリの構造を保ったまま全てコピーしなさい。

コピー後に tree コマンド

```
t190900@s01cd0542-160:~$ tree ~/kiso2-2019/texmf
```

の出力結果をテキストファイル ex01-2.txt に保存し、kiso2 コマンドを用いて提出しなさい。

コピー後の tree コマンドの出力例は以下の通りである。

```
t190900@s01cd0542-160:~/kiso2-2018 $ tree ~/kiso2-2019/texmf
/home/t190900/kiso2-2019/texmf
texmf
├── dvipdfmx
├── dvips
│   └── config
├── tex
│   └── generic
│       └── config
├── texdoctk
│   └── texdocrc.defaults
├── texmf.d
│   └── 00debian.cnf
└── web2c
    ├── mktex.cnf
    └── texmf.cnf

9 directories, 4 files
```

ディレクトリを、その下にあるファイルを含めて全てコピーしたい場合は、ファイルをコピーするコマンド*cp* のオプション *cp -r* を利用することができます。 使い方は、“ シェル ディレクトリ コピー” などのキーワードで調べてみましょう。

シェル上で、あるディレクトリにある**すべてのファイル**を表現するためには * (アスタリスク、ほし) を使うことができます。 たとえば、*rm * はカレントディレクトリにある**すべてのファイル**が消去されます。*cp * dir* は、カレントディレクトリにある**すべてのファイル**が dir ディレクトリにコピーされます。また、*記号はコマンドの一部に使うこともできます。たとえば、*ls *.c* というコマンドは、カレントディレクトリにあるファイル名が□**.c**□で終わる**すべてのファイル**を表示します。

演習課題 ex01-3.txt

ターミナルを用いて、次の文章が表すディレクトリとファイルを作成しなさい。

ホームディレクトリの下に kiso2-2019 ディレクトリが存在し、その kiso2-2019 ディレクトリに下には2つのディレクトリ ex00 と ex01 のみが存在している。
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。
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。

これらを作成した後に、カレントディレクトリを **~/kiso2-2019** に移動し（*pwd コマンドを実行したときに/home/自分の学籍番号/kiso2-2019/と表示される状態で*） 以下の2つのコマンドを実行しなさい。

```
t190900@s01cd0542-160:~/kiso2-2019$ cat ex00/foo
    □
 (実行結果)
t190900@s01cd0542-160:~/kiso2-2019$ tree
    :
 (実行結果)
```

これらのコマンドで表示された内容をテキストファイル **ex01-3.txt** に保存し、kiso2 コマンドを用いて提出しなさい。 ただし、プロンプトや入力したコマンドの表記については、提出するテキストファイルの内容に含んでも、含まなくてもどちらでもよい。

この課題では□□kiso2-2019 ディレクトリに下には2つのディレクトリ ex00 と ex01 □のみ』が存在している」と指示されています。 演習課題 ex01-1.sh や ex02-2.txt を実行した場合□~/kiso2-2019 ディレクトリの下にはすでに幾つかのファイルやディレクトリが作成されているかもしれません。 そのような場合は、それらのファイルやディレクトリを消去して、この課題の指示通りのディレクトリ構成となるようにして下さい。

演習課題 ex01-5.sh

演習課題 **ex01-1.sh** では、演習内容を実現する複数のシェルコマンドを一行ずつ並べたテキストファイル**ex01-1.sh**を作成した。 このように、シェルコマンドを実行する順番に記述したテキストファイルは**シェルスクリプト** と呼ばれ、プログラムとして実行することができる。 (中身はただのテキストファイルだが、文章ファイルなどと区別するためにファイル名には慣習的に **.sh**というファイル拡張子が使われる。)

たとえば、以下のような内容のテキストファイル**ex01-5-sample.sh**は、

ex01-5-sample.sh

```
mkdir ~/kiso2-2019/ex01-sample
echo "~/kiso2-2019/ex01-sample を作成しました。"
```

```
cp /etc/services ~/kiso2-2019/ex01-sample/.
echo "~/kiso2-2019/ex01-sample に /etc/services をコピーしました。"
```

次のように**bash**というコマンド(シェル実行ファイル)を使って実行することができ、ファイルに書かれた順番にそれらのシェルコマンドが実行される。

```
t190900@s01cd0542-160:~/kiso2-2019$ tree
.
└── ex01-5-sample.sh

0 directories, 1 file
t190900@s01cd0542-160:~/kiso2-2019$ bash ex01-5-sample.sh
~/kiso2-2019/ex01-sample を作成しました。
~/kiso2-2019/ex01-sample に /etc/services をコピーしました。
t190900@s01cd0542-160:~/kiso2-2019$ tree
.
├── ex01-5-sample.sh
└── ex01-sample
    └── services

1 directory, 2 files
```

さて、このようなシェルスクリプトとして、適当なディレクトリ名を与えると自動的に **~/kiso2-2019/** ディレクトリの下に与えられた名前のディレクトリを作成するシェルスクリプト **ex01-5.sh**を作成したい。たとえば、以下のようにファイル名の後ろに**ex02** という名前を与えて実行すると、**ex01-5.sh**は、

```
t190900@s01cd0542-160:~/kiso2-2019$ tree
.
├── ex01-5-sample.sh
└── ex01-sample
    └── services

1 directory, 2 files
t190900@s01cd0542-160:~/kiso2-2019$ bash ex01-5.sh ex02
~/kiso2-2019/ex02 を作成しました。
t190900@s01cd0542-160:~/kiso2-2019$ tree
.
├── ex01-5-sample.sh
├── ex01-sample
│   └── services
└── ex02

2 directories, 2 files
```

のようなメッセージとともに新しいディレクトリ **~/kiso2-2019/ex02** を作成する。

このようなシェルスクリプト **ex01-5.sh** を作成し、kiso2コマンドを用いて提出しなさい。

実行例 *bash ex01-5.sh ex02* での *ex02* のように、シェルスクリプトなどのプログラム実行時にコマンドの後ろに空白で区切られて与えられる文字列を**コマンドラインオプション**と呼ぶ。 たとえば、**removeme.txt** というファイルを削除するためのコマンド *rm removeme.txt* では、**removeme.txt** が rm コマンドのコマンドラインオプションである。

このようなコマンドラインオプションで指定された文字列をシェルスクリプトの中で利用する場合、**$1** のような特殊な変数によってコマンドラインオプションの文字列を使うことができる。

たとえば、次のようなecho コマンド1行だけからなるシェルスクリプトは、

option.sh

```
echo $1
```

次のように実行される。

```
t190900@s01cd0542-160:~/kiso2-2019$ bash option.sh Hello
Hello
t190900@s01cd0542-160:~/kiso2-2019$ bash option.sh あいうえお
あいうえお
```

もちろん、**$2**, **$3** も、さらには **$0** もある。 これらの変数がどのように働くのか、自分で調べて実験してみて下さい。