情報システム・セキュリティ実験 I

課題2 2021/4/20 19T325 樋口史弥

1

1-1

1-1-a

lsから始まるコマンドを列挙する

```
$ ls\t\t
ls            lsblk         lscpu         lsinitramfs   lslocks       lsmem
lsns          lspci         lsusb
lsattr        lsb_release   lshw          lsipc         lslogins      lsmod
lsof          lspgpot       lsvmbus
```

1-1-b

カレントディレクトリにある、ファイルとディレクトリを列挙する。
表示されるものは、「.」から始まる物も表示される。

```
$ ls␣\t\t
.bash_history   .cache/         experiment/     .npm/           server.conf
.vim/
.bash_logout    .config/        go/             .profile        .ssh/
.viminfo
.bashrc         Documents/      .mysql_history  React/          udo sy
```

1-1-c

mvを押した後、Tabキーを押すとスペースが一つ空き、mvコマンドに確定する。
cpを押した後、Tabキーを押すとcpから始まるコマンドの候補が列挙される。

```
$ mv\t
$ mv␣
$ cp\t
cp                        cpgr                      cpp-9
cpan                      cpio                      cppw
cpan5.30-x86_64-linux-gnu  cpp                      cpupower
```

1-2

- pwd
  カレントディレクトリの絶対パスを表示する。
- mkdir dir1
  dir1というディレクトリを作成する。-pオプションをつけると必要に応じて親ディレクトリも作成する
-

系列１

```
$ pwd
/home/vagrant/experiment
$ cd
$ pwd
/home/vagrant
$ mkdir tmpdir
$ ls
Documents  experiment  go  React  tmpdir
$ cp ~/experiment/Exp1-sample/f1.txt ~/tmpdir/f1.txt
$ cp ~/experiment/Exp1-sample/f2.txt ~/tmpdir/
$ mkdir tmpdir/sd1 tmpdir/sd2
$ cp ~/experiment/Exp1-sample/sd1/f3.txt ~/experiment/Exp1-
sample/sd1/f4.txt ~/tmpdir/sd1
$ cd tmpdir/sd2
$ cp ~/experiment/Exp1-sample/sd2/f5.txt .
$ cp ~/experiment/Exp1-sample/sd2/ssd/*.txt ssd
$ cd ../..
$ ls -R tmpdir/
tmpdir/:
f1.txt  f2.txt  sd1  sd2

tmpdir/sd1:
f3.txt  f4.txt

tmpdir/sd2:
f5.txt  ssd

tmpdir/sd2/ssd:
f6.txt  f7.txt
$ rm -rf tmpdir/
```

系列2

```
$ cd
$ cp -r ~/experiment/Exp1-sample/ tmpdir
$ ls -R tmpdir
tmpdir:
f1.txt  f2.txt  sd1  sd2

tmpdir/sd1:
f3.txt  f4.txt
```

```
tmpdir/sd2:
f5.txt  ssd

tmpdir/sd2/ssd:
f6.txt  f7.txt
$ rm -rf tmpdir
```

系列3

```
$ cd
$ mkdir tmpdir tmpdir/sd
$ cp ~/experiment/Exp1-sample/f1.txt tmpdir/new1.txt
$ ls -R
.:
new1.txt  sd

./sd:
$  mv new1.txt new2.txt
$ ls -R
.:
new2.txt  sd

./sd:
$ mv new2.txt sd/new3.txt
$ ls -R
.:
sd

./sd:
new3.txt
$ rmdir sd
rmdir: failed to remove 'sd': Directory not empty
$ mv sd/* .
$ ls -R
.:
new3.txt  sd

./sd:
$ rmdir sd
$ ls -R
.:
new3.txt
$ cd ~; rm -rf tmpdir
```

1-3

```
$ mkdir 19 20
$ mkdir 19/exp1 19/inet1 20/inet1 20/inet2
```

```
$ mv exp1/19/a1 exp1/19/a2 19/exp1/
$ mv inet1/A1 inet1/19/b1 inet1/19/b2 19/inet1/
$ cp 19/inet1/A1 inet1/20/c1 inet1/20/c2 20/inet1/
$ mv inet2/20/d1 inet2/20/d2 20/inet2/
$ rm -r exp1/ inet1/ inet2/
```

2

2-1

```
$ ls -t
b.txt  a.txt  c.txt
```

2-2

```
$ ls -a
.  ..  a.txt  b.txt  c.txt
```

2-3

```
$ ls -tr
c.txt  a.txt  b.txt
```

2-4

```
$ cp -p a.txt tmp.txt
$ ll
total 24
drwxr-xr-x 2 vagrant vagrant 4096 Apr 20 07:29 ./
drwxrwxr-x 5 vagrant vagrant 4096 Apr 20 04:06 ../
-rw-r--r-- 1 vagrant vagrant   28 May 25  2016 a.txt
-rw-r--r-- 1 vagrant vagrant   32 Apr  6  2017 b.txt
-rw-r--r-- 1 vagrant vagrant   34 Apr 11  2015 c.txt
-rw-r--r-- 1 vagrant vagrant   28 May 25  2016 tmp.txt
```

2-5

```
$ mkdir -p a/b
$ ls
a  a.txt  b.txt  c.txt  tmp.txt
```

```
$ ls a
b
```

2-6

- cp
  - -i
    上書きする前に確認をする
  - -r
    コピー元のディレクトリを指定した場合、再帰的にコピーする
  - -a
    サブディレクトリや属性なども含め可能な限りすべてを保持しながらコピーする
- mv
  - -f
    移動先に同名ファイルがあっても確認せずに上書きする

3

3-1

```
#include <stdio.h>
#define DEFAULT_LOOP 3 // The number of loops when the entered loop range
is incorrect
#define MAX_LOOP 5 // The maximum number of loops
main()
{
        int loopnum;
        int loop;
        printf("Input #loops (between 1 and %d): ", MAX_LOOP);
        scanf("%d", &loopnum);
        loopnum = ((loopnum <= 0) || (loopnum > MAX_LOOP)) ? DEFAULT_LOOP :
loopnum;
```

```
include <stdio.h>
#define DEFAULT_LOOP 3 // The number of loops when the entered loop range
is incorrect
#define MAX_LOOP 5 // The maximum number of loops
main()
{
        int loopnum;
        int loop;
        printf("Input #loops (between 1 and %d): ", MAX_LOOP);
        scanf("%d", &loopnum);
        loopnum = ((loopnum <= 0) || (loopnum > MAX_LOOP)) ? DEFAULT_LOOP :
loopnum;
        printf("#loops (stdout): %d\n", loopnum); // Output to stdout
        fprintf(stderr, "#loops (stderr): %d\n", loopnum); // Output to
```

```
stderr
        for (loop = 1; loop <= loopnum; loop++) {
                printf("(stdout) %d\n", loop); // Output to stdout
                fprintf(stderr, "(stderr) %d\n", loop); // Output to stderr
        }
}
```

3-2

3-2-a

```
/loopnum\t
```

3-2-b

```
:%s/loop/lp/g
```

3-3

```
include <stdio.h>
#define DEFAULT_LOOP 3 // The number of lps when the entered lp range is
incorrect
#define MAX_LOOP 5 // The maximum number of lps
main()
{
        int lpnum;
        int lp;
        printf("Input #lps (between 1 and %d): ", MAX_LOOP);
        fflush(stdout);
        scanf("%d", &lpnum);
        lpnum = ((lpnum <= 0) || (lpnum > MAX_LOOP)) ? DEFAULT_LOOP :
lpnum;
        printf("#lps (stdout): %d\n", lpnum); // Output to stdout
        fprintf(stderr, "#lps (stderr): %d\n", lpnum); // Output to stderr
        fflush(stdout);
        for (lp = 1; lp <= lpnum; lp++) {
                printf("(stdout) %d\n", lp); // Output to stdout
                fflush(stdout);
                fprintf(stderr, "(stderr) %d\n", lp); // Output to stderr
        }
}
```