

# Linux Command 入門 2

Secprj Intro-phase

標準入力 / 標準出力

standard input / standard output

標準入力 / 標準出力

standard input / standard output

# 入力と出力 input / output

コマンドには入力と出力となるファイルが存在する

\$ cp の場合は...

コマンド本体

引数

\$ cp file1 file2

# 入力と出力 input / output

コマンドには入力と出力となるファイルが存在する

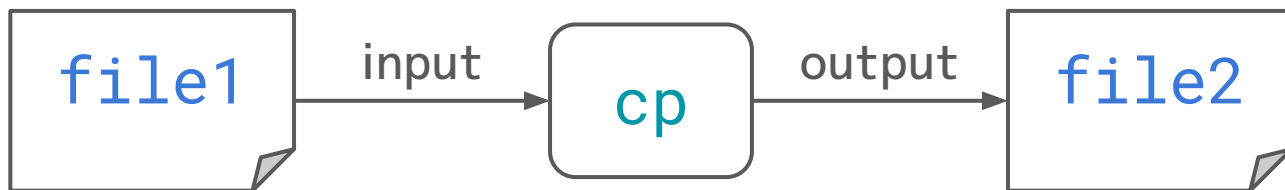
\$ cp の場合は...

	コマンド本体	引数	
\$	cp	file1	file2
		input	output

# 入力と出力 input / output

コマンドには入力と出力となるファイルが存在する

\$ cp の場合は...



- inputされた  
ファイルを複製
- 指定された名前  
でoutput

```
vagrant@ubuntu-focal:~$ ls
```

```
hoge
```

```
vagrant@ubuntu-focal:~$ cp hoge fuga
```

```
vagrant@ubuntu-focal:~$ ls
```

```
fuga  hoge
```

```
vagrant@ubuntu-focal:~$ █
```

## Terminalへの出力がある

```
vagrant@ubuntu-focal:~$ ls
```

```
hoge
```

```
vagrant@ubuntu-focal:~$ cp hoge fuga
```

```
vagrant@ubuntu-focal:~$ ls
```

```
fuga  hoge
```

```
vagrant@ubuntu-focal:~$
```



## Terminalへの出力がない

```
vagrant@ubuntu-focal:~$ ls
```

```
hoge
```

```
vagrant@ubuntu-focal:~$ cp hoge fuga
```

```
vagrant@ubuntu-focal:~$ ls
```

```
fuga  hoge
```

```
vagrant@ubuntu-focal:~$
```

標準入力 / 標準出力

standard input / standard output

\$ ls は出力先を指定していない

```
vagrant@ubuntu-focal:~$ ls
```

```
hoge
```

```
vagrant@ubuntu-focal:~$ cp hoge fuga
```

```
vagrant@ubuntu-focal:~$ ls
```

```
fuga  hoge
```

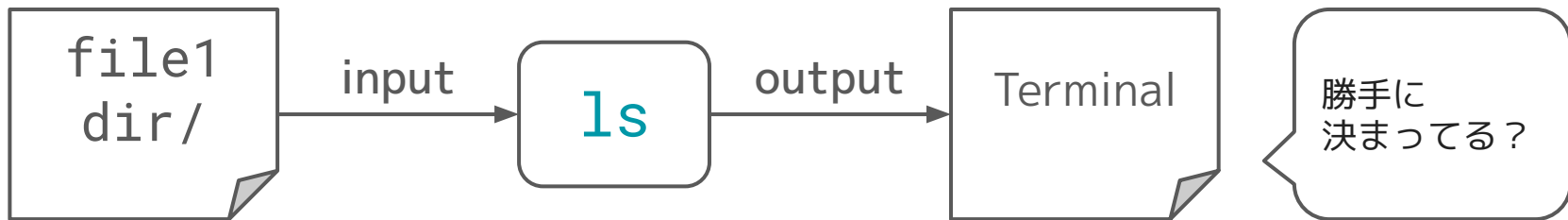
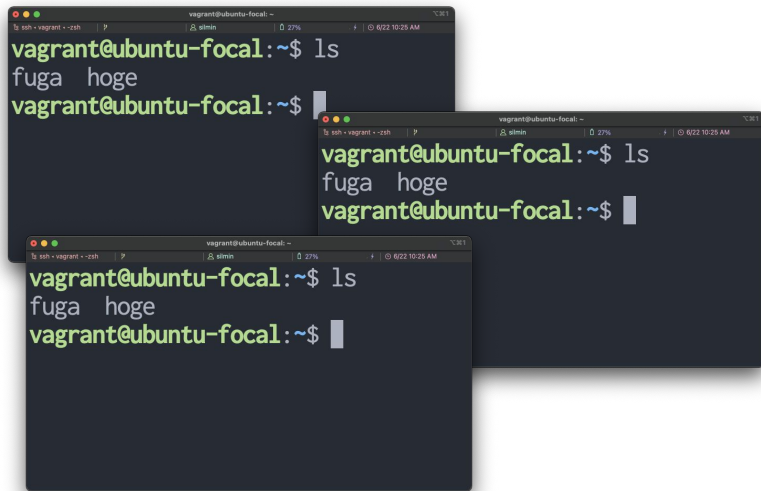
```
vagrant@ubuntu-focal:~$
```

# どうやって出力先を決めるか

\$ cp のように指定したわけではない

Terminalがたくさんあったら？

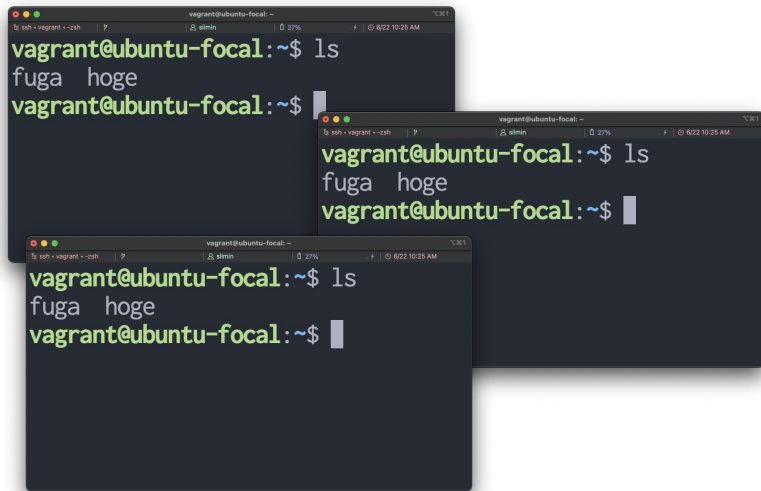
指定してないけど，出力先が決まっている



# どうやって出力先を決めるか

何も指定しない場合に暗黙的に使用される  
出力先のことを標準出力という

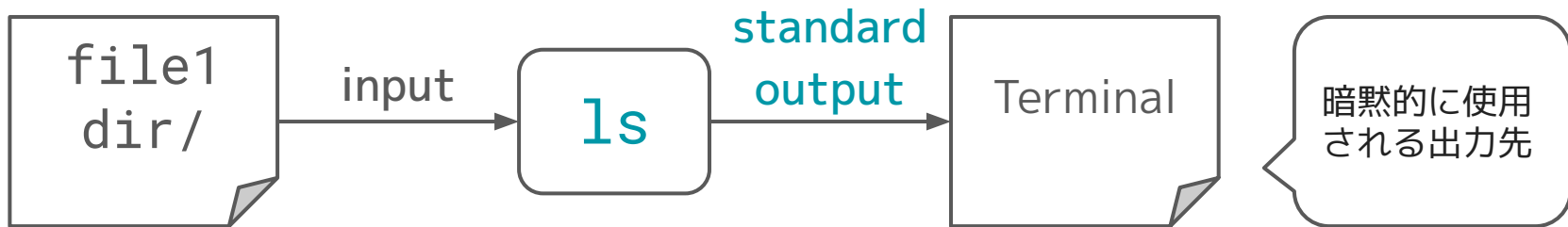
(厳密にはlsが標準出力を指定している)



```
vagrant@ubuntu-focal:~$ ls
fuga hoge
vagrant@ubuntu-focal:~$
```

```
vagrant@ubuntu-focal:~$ ls
fuga hoge
vagrant@ubuntu-focal:~$
```

```
vagrant@ubuntu-focal:~$ ls
fuga hoge
vagrant@ubuntu-focal:~$
```



標準入力 / 標準出力

standard input / standard output

# 標準入力は標準出力の入力版

何も指定しないとき、コマンドが入力を求めた場合、標準入力から入力する

標準入力

```
vagrant@ubuntu-focal:~$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006,
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
1+1
2
10-2
8
vagrant@ubuntu-focal:~$
```

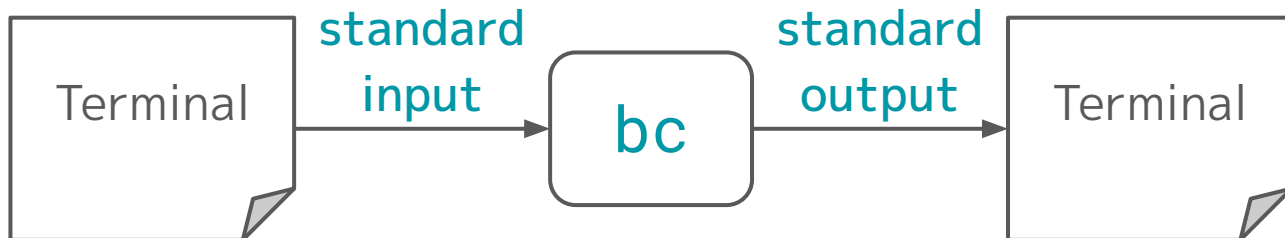
bcコマンドは入力された文字列に対し、簡単な四則演算をして出力する

# 標準入力は標準出力の入力版

何も指定しないとき、コマンドが入力を求めた場合、標準入力から入力する

標準入力

```
vagrant@ubuntu-focal:~$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006,
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
1+1
2
10-2
8
vagrant@ubuntu-focal:~$
```





# パイプとリダイレクト

どちらも標準入出力を引き回す方法

パイプ `|` : コマンドの標準出力を他のコマンドの標準入力にする

リダイレクト `<`, `>`, ... : コマンドの標準入出力を他のファイル宛にする

```
$ ls -l /bin | less
```

```
$ bc < file1 > file2
```

# パイプ (Pipe)

コマンドの標準出力を他のコマンドの標準入力にする

lsの引数は何でもいい  
今回はlessを使う関係上  
長めなやつを選んだ

```
$ ls -l /usr/bin
```

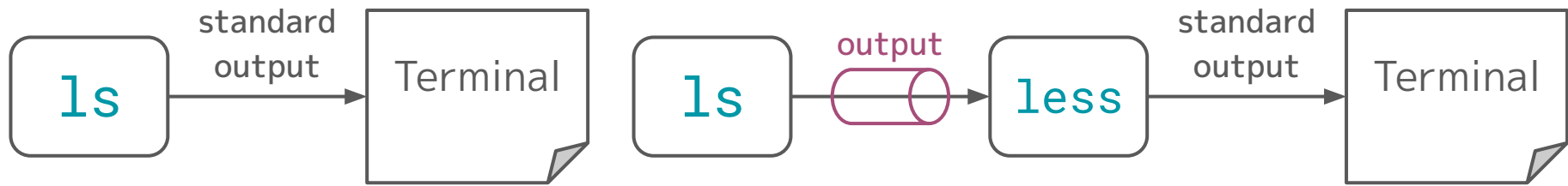
```
$ ls -l /usr/bin | less
```

# パイプ (Pipe)

コマンドの標準出力を他のコマンドの標準入力にする

```
$ ls -l /usr/bin
```

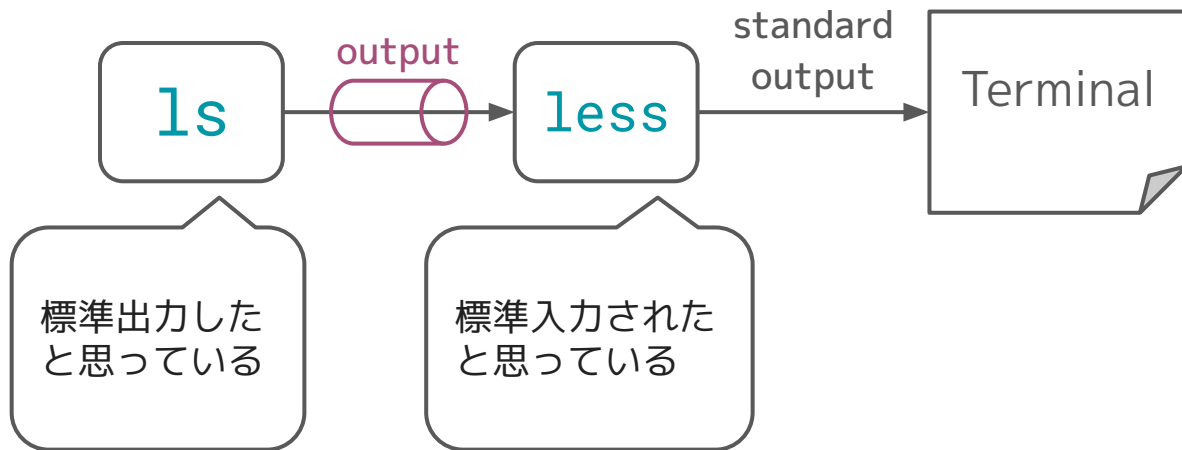
```
$ ls -l /usr/bin | less
```



# パイプ (Pipe)

コマンドの標準出力を他のコマンドの標準入力にする

```
$ ls -l /usr/bin | less
```



# リダイレクト (Redirect)

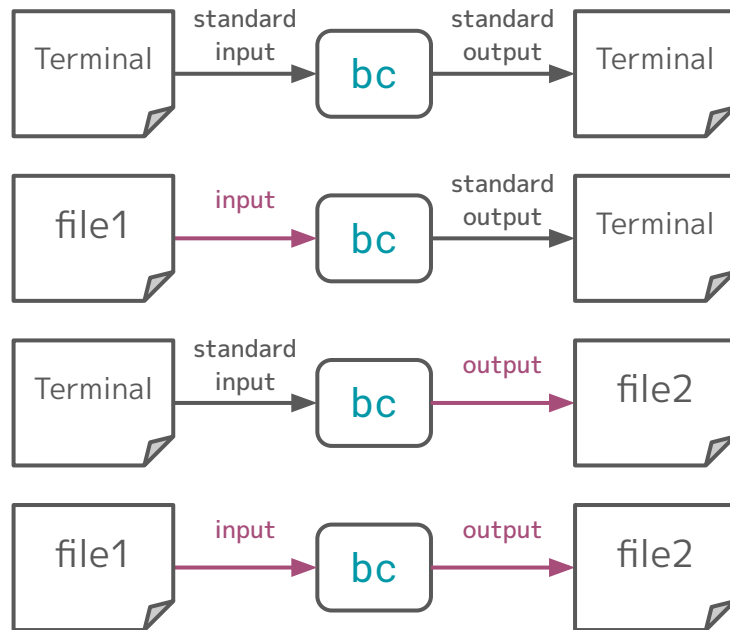
コマンドの標準入出力を他のファイル宛にする

```
$ bc
```

```
$ bc < file1
```

```
$ bc > file2
```

```
$ bc < file1 > file2
```



# パイプとgrepコマンド

grepコマンドは検索文字列と検索対象を受け取り，結果を出力する

```
$ grep str file1
```

```
vagrant@ubuntu-focal:~$ cat foo.txt
hoge1
foo1
foo2
hoge2
bar1
foo3
fuga1
hoge3
vagrant@ubuntu-focal:~$ grep hoge foo.txt
hoge1
hoge2
hoge3
vagrant@ubuntu-focal:~$
```

# パイプとgrepコマンド

検索対象は標準入力から受け取ることもできる

```
$ ls -l /usr/bin | grep apt
```

```
vagrant@ubuntu-focal:~$ ls -l /usr/bin | grep apt
-rwxr-xr-x 1 root root 7415 Apr 30 18:13 add-apt-repository
-rwxr-xr-x 1 root root 18824 Mar 12 11:47 apt
lrwxrwxrwx 1 root root 18 Apr 30 18:13 apt-add-repository -> add-apt-repository
-rwxr-xr-x 1 root root 88536 Mar 12 11:47 apt-cache
-rwxr-xr-x 1 root root 31192 Mar 12 11:47 apt-cdrom
-rwxr-xr-x 1 root root 27016 Mar 12 11:47 apt-config
-rwxr-xr-x 1 root root 27104 Mar 12 11:47 apt-extracttemplates
-rwxr-xr-x 1 root root 281056 Mar 12 11:47 apt-ftparchive
-rwxr-xr-x 1 root root 47576 Mar 12 11:47 apt-get
-rwxr-xr-x 1 root root 27931 Mar 12 11:47 apt-key
-rwxr-xr-x 1 root root 63960 Mar 12 11:47 apt-mark
-rwxr-xr-x 1 root root 47504 Mar 12 11:47 apt-sortpkgs
lrwxrwxrwx 1 root root 3 Feb 26 2020 captoinfo -> tic
-rwxr-xr-x 1 root root 11541 Aug 3 2019 debconf-apt-progress
vagrant@ubuntu-focal:~$
```

## パイプとリダイレクトを組み合わせた例

\$ **cat** hoge.out | **grep** hoge.9 > hoge9

```
vagrant@ubuntu-focal:~$ tail hoge.out  
hoge.91  
hoge.92  
hoge.93  
hoge.94  
hoge.95  
hoge.96  
hoge.97  
hoge.98  
hoge.99  
hoge.100
```

\$ **tail** はファイルの最後  
数行のみを出力するコマンド

```
vagrant@ubuntu-focal:~$ cat hoge.out | grep hoge.9 > hoge9  
vagrant@ubuntu-focal:~$ cat hoge9  
hoge.9  
hoge.9  
hoge.9  
hoge.90  
hoge.91  
hoge.92  
hoge.93  
hoge.94  
hoge.95  
hoge.96  
hoge.97  
hoge.98  
hoge.99  
vagrant@ubuntu-focal:~$
```



# 正規表現 (regular expression)

文字列が特定のパターンにマッチするかどうかを判定してくれる

このほかにもたくさんある

文字	説明
.	任意の1文字にマッチ
+	直前の文字が1回以上繰り返す
*	直前の文字が0回以上繰り返す
\n	改行文字
\t	タブ文字

# 正規表現 (regular expression)

CLI操作をする上で便利な場面がいくつかある

\$ **ls** -l \*.py

```
vagrant@ubuntu-focal:~$ ls -l
total 12
-rw-rw-r-- 1 vagrant vagrant    0 Jun 22 09:23 foo.py
-rw-rw-r-- 1 vagrant vagrant   44 Jun 22 06:55 foo.txt
-rw-rw-r-- 1 vagrant vagrant    0 Jun 22 09:23 fuga.txt
-rw-rw-r-- 1 vagrant vagrant 1105 Jun 22 03:22 hoge.out
-rwxrwxr-x 1 vagrant vagrant   77 Jun 22 05:54 hoge.sh
-rw-rw-r-- 1 vagrant vagrant    0 Jun 22 09:23 hoge.txt
-rw-rw-r-- 1 vagrant vagrant    0 Jun 22 09:23 piyo.py
vagrant@ubuntu-focal:~$ ls -l *.py
-rw-rw-r-- 1 vagrant vagrant 0 Jun 22 09:23 foo.py
-rw-rw-r-- 1 vagrant vagrant 0 Jun 22 09:23 piyo.py
vagrant@ubuntu-focal:~$
```