



# Programming Languages Linux Commands #2



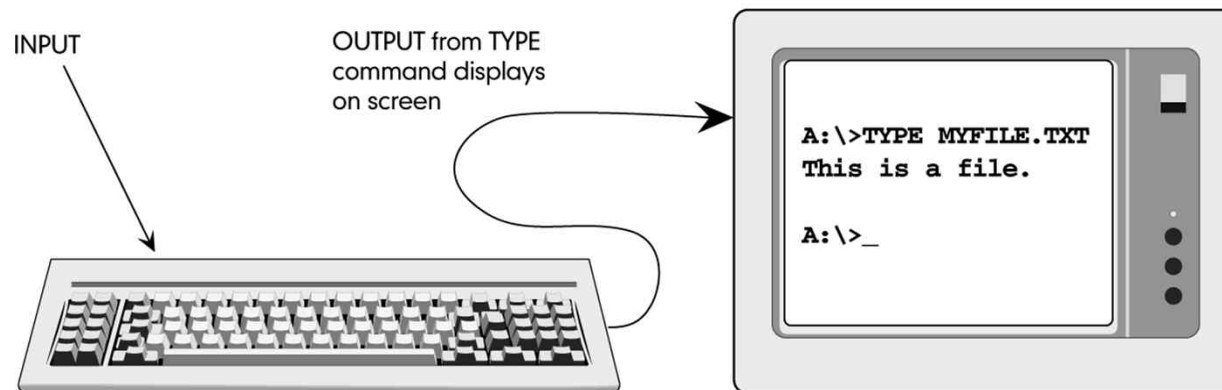
남 범 석

bnam@skku.edu



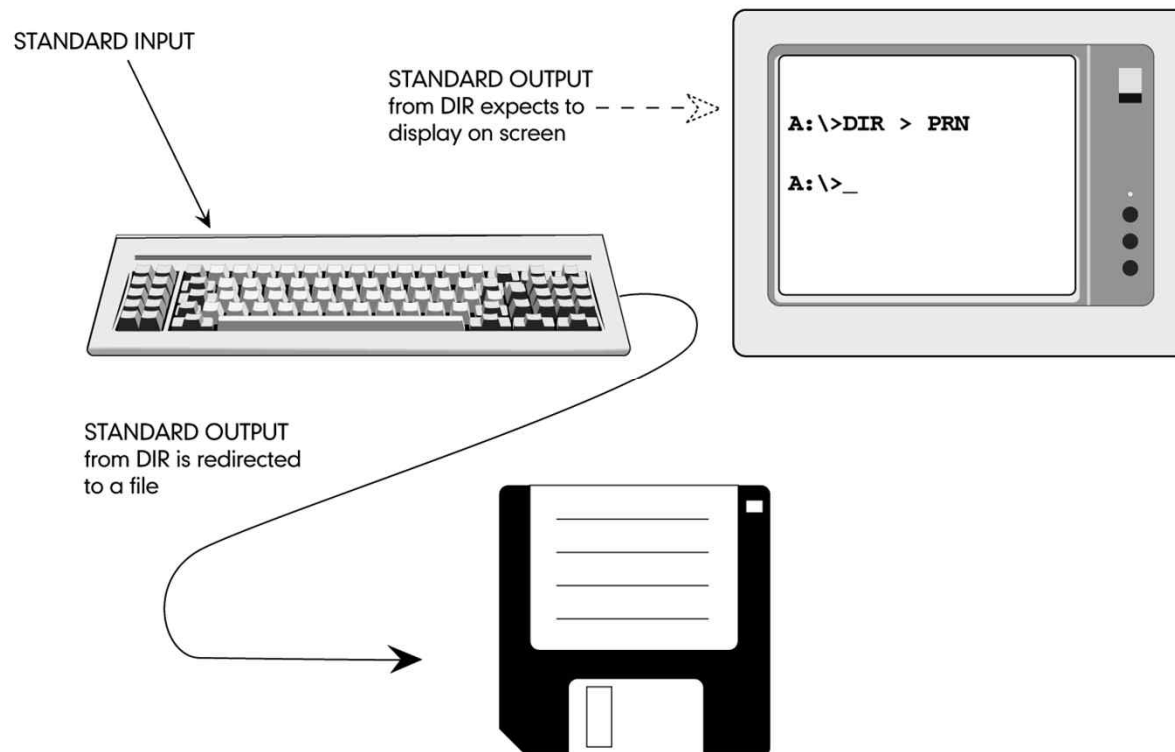
## I/O redirection

- There are always three default *files* open
  - stdin (the keyboard) : file descriptor 0
  - stdout (the screen) : file descriptor 1
  - stderr (error messages output to the screen) : file descr. 2



## I/O redirection

- These, and any other open files, can be *redirected*.
- Redirection simply means capturing output from a file, or program and sending it as input to another file, command or program



## Input redirection

- `command < filename` (or `command 0< filename`)
  - read input from a file

```
bsnam@dicl1:~$ cat test.c
#include <stdio.h>
#include <limits.h>

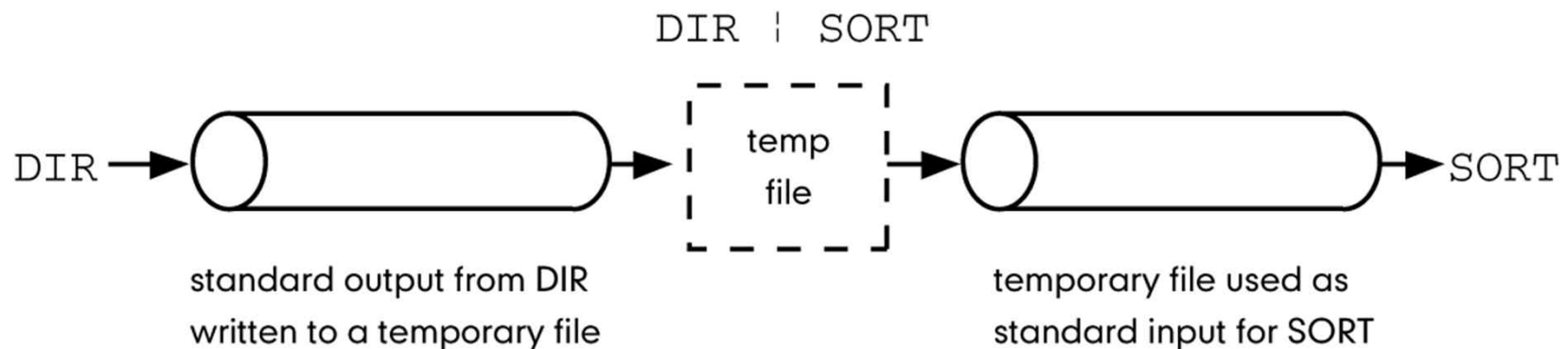
int main()
{
    char buf[1024];
    scanf("%s", buf);
    printf(">> %s\n", buf);
}
bsnam@dicl1:~$ gcc test.c
bsnam@dicl1:~$ ./a.out < test.c
>> #include
bsnam@dicl1:~$
```

# Output redirection

- **> filename (or |>filename)**
  - # Redirect stdout to file "filename."
- **>> filename (or |>>filename)**
  - # Redirect and append stdout to file "filename."
- **2>filename**
  - # Redirect stderr to file "filename."
- **2>>filename**
  - # Redirect and append stderr to file "filename."
- **&>filename**
  - # Redirect both stdout and stderr to file "filename."
- **M>N**
  - # "M" is a file descriptor, which defaults to 1, if not explicitly set.
  - # "N" is a filename.
- **M>&N**
  - # "M" is a file descriptor, which defaults to 1, if not set.
  - # "N" is another file descriptor. (1: stdout, 2: stderr)

# Pipe

- Standard output of one program used as standard input to next program
  - Pipe symbol is the vertical broken bar | and is used between two commands.



## tee

- Reads standard input and writes it to both standard output and one or more files, effectively duplicating its input

