



Shell Programming (additional)

Shell Scripts

❑ A collection of commands

- Ex:

```
#!/bin/sh
```

```
ls -al
```

```
touch aa
```

```
cp aa bb
```

❑ What you have to learn?

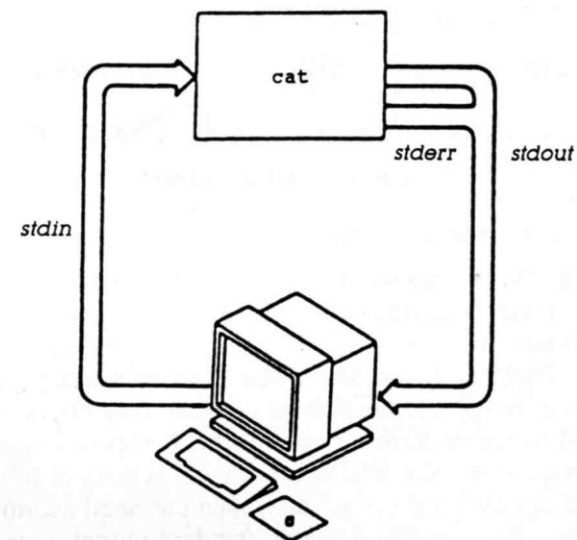
- Some magic in UNIX environment
- UNIX commands
- Shell program structure

Input/Output Redirection (1)

- ❑ Every process has 3 default file descriptors

Name	I/O	Descriptor #
<i>stdin</i>	input	0
<i>stdout</i>	output	1
<i>stderr</i>	error output	2
User-defined	Input/output	3 ~ 19

- ❑ In normal situation
 - *stdout* and *stderr* print on the terminal
 - *stdin* reads input from keyboard



Input/Output Redirection (2)

❑ Redirection

- Change the direction of stdin, stdout, stderr, or any other user-defined file descriptor
 - Create files – Save the output of command in the file
 - Append to files – Append the output of command in the file
 - Use existing file as input – Make the command reads stdin from the file
 - Merge two output streams
 - Use part of the Shell command as input

Input/Output Redirection (3)

Method	Description
<code>cmd < file</code>	Open the file as stdin of cmd
<code>cmd > file</code>	Write stdout of cmd in the following file (noclubber)
<code>cmd >> file</code>	Append stdout of cmd to the following file
<code>N>&M</code>	Merge file descriptor N to file descriptor M
<code>cmd1 cmd2</code>	Pipe stdout of cmd1 into stdin of cmd2
<code>cmd << del</code>	Take stdin from here, up to the delimiter del
<code>n>&-</code>	Close file descriptor

- ❑ “Redirection” in `sh(1)`, or “Input/Output” in `tcsh(1)`

Input/Output Redirection (4)

□ Examples

- `% echo "we have several shell" > chapter1`
- `% sed -e "s/shell/SHELL/g" < chapter1`
 - we have several SHELL
- `% sed -e "s/SHELL/shell/g" < chapter1 > newchapter1`
 - stdout goes to newchapter1 file
 - stderr still goes to terminal



- `% sed -e "s/SHELL/shell/g" < chapter1 > newchapter1 2> errchapter`
 - stdout goes to newchapter1 and stderr goes to errchapter



- `% sed -e "s/SHELL/shell/g" < chapter1 > newchapter1 >& errchapter`



- `% sed -e "s/SHELL/shell/g" < chapter1 > newchapter1 2>&1`
 - Both stdout and stderr go to newchapter1

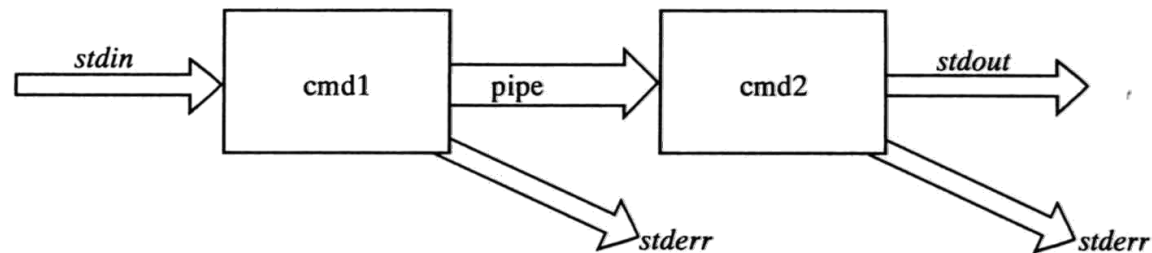


- `% sed -e "s/SHELL/shell/g" < chapter1 >& newchapter1`

Input/Output Redirection (5)

□ pipe

- Connect the stdout of one command to the stdin of another
- Two commands will operate asynchronously



□ Example

- `% dmesg | grep CPU | less`
- To merge stderr with stdout and pipe to next command



➤ `% command arguments 2>&1 | nextcommand`



➤ `% command arguments |& nextcommand`



- `% exec 4>&- # close file descriptor 4`
- `% exec 1>&- # close stdout`

Arrays

❑ Use "eval"

```
#!/bin/sh -x

N=0
while [ $N -lt 10 ]
do
    eval number$N=$N
    N=$((N + 1))
done

N=8
eval echo "The number is \${number$N}."

echo "The number5 is $number5."
```