Systems Programming

Bonus Assignment

Haehyun Cho

School of Software
Soongsil University

Cyber Security
Research Center

A basic multi-process program

- This assignment is to implement a basic multi-process program.

- Your program MUST take a path of a text file as an argument.
  - thus, you should run your program like
    ```
    ./yourprogram path-to-textfile
    ```

- Your program MUST read a text file.
  - the text file has multiple lines more than 5 lines of text data

- Your program MUST read the file by using 5 processes as follows

PIDs

1
Read a line of the text file
Print the line with its PID
Let the next process read a line

2
Read a line of the text file
Print the line with its PID
Let the next process read a line

3
Read a line of the text file
Print the line with its PID
Let the next process read a line

4
Read a line of the text file
Print the line with its PID
Let the next process read a line

5
Read a line of the text file
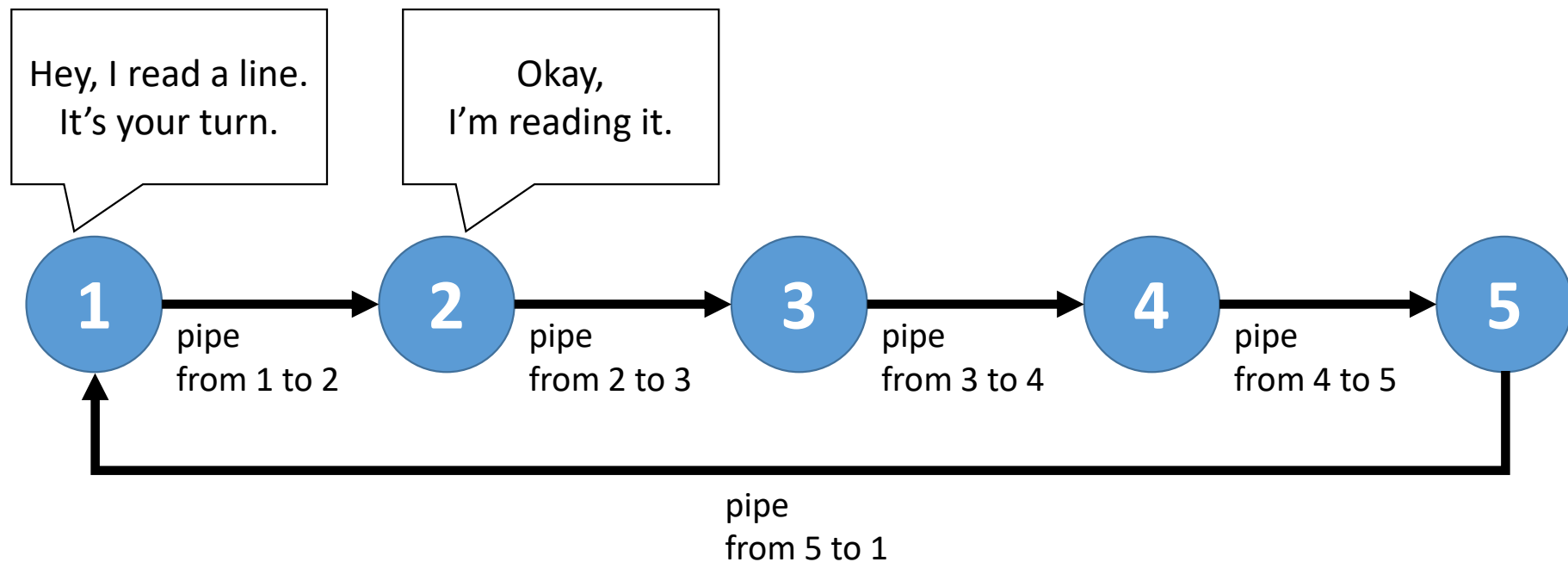Print the line with its PID
Let the next process read a line

Text file

1 first
2 second
3 third
4 fourth
5 fifth
1 sixth
2 seventh
3 eighth
4 nineth
5 tenth
1 eleventh
2 twelveth
3 thirteenth
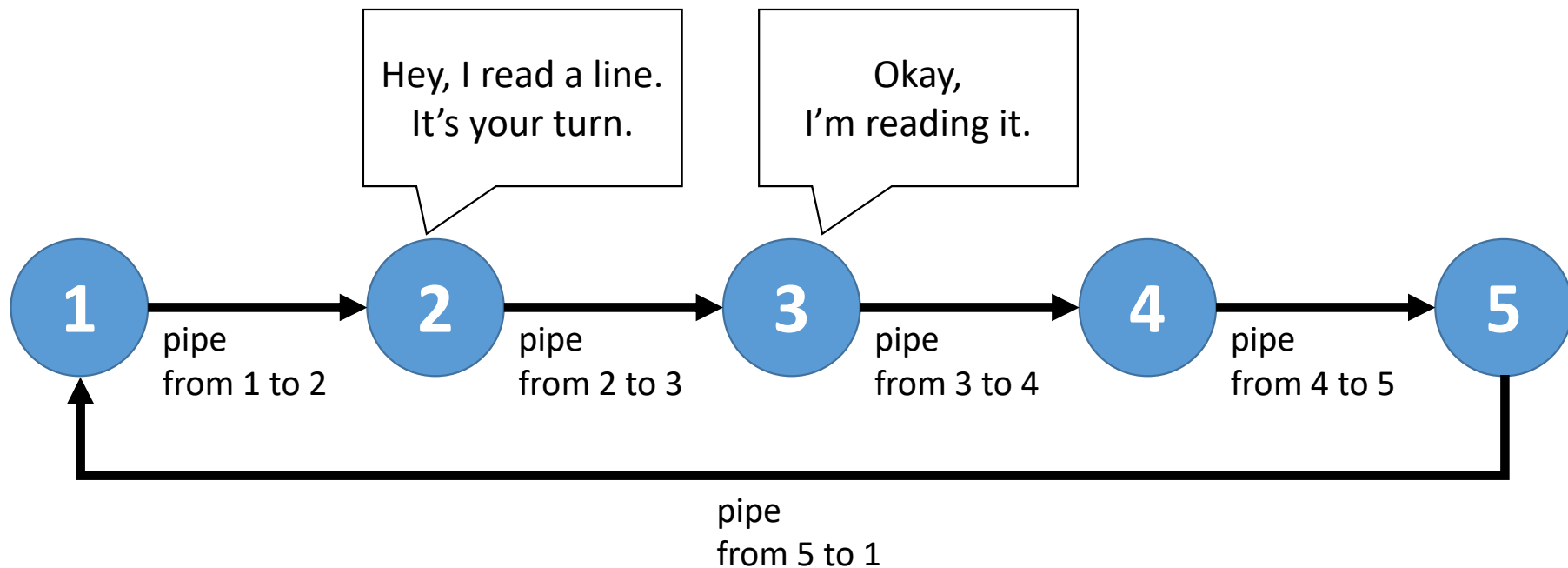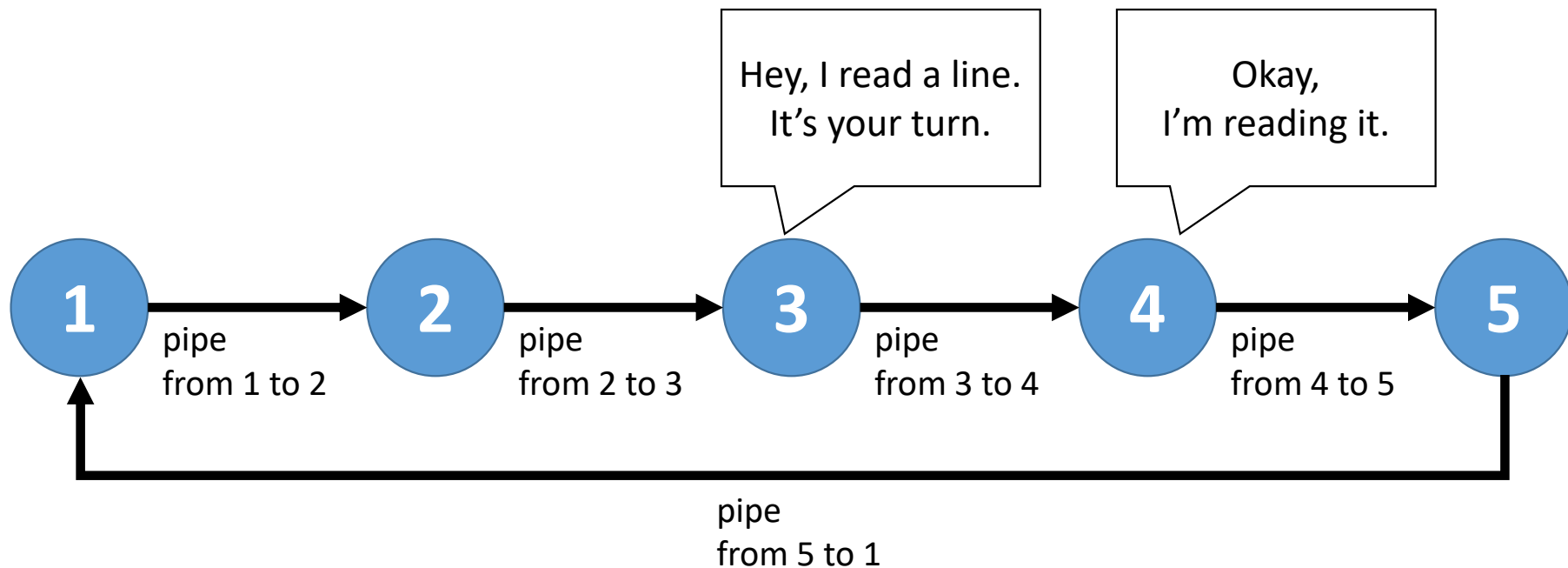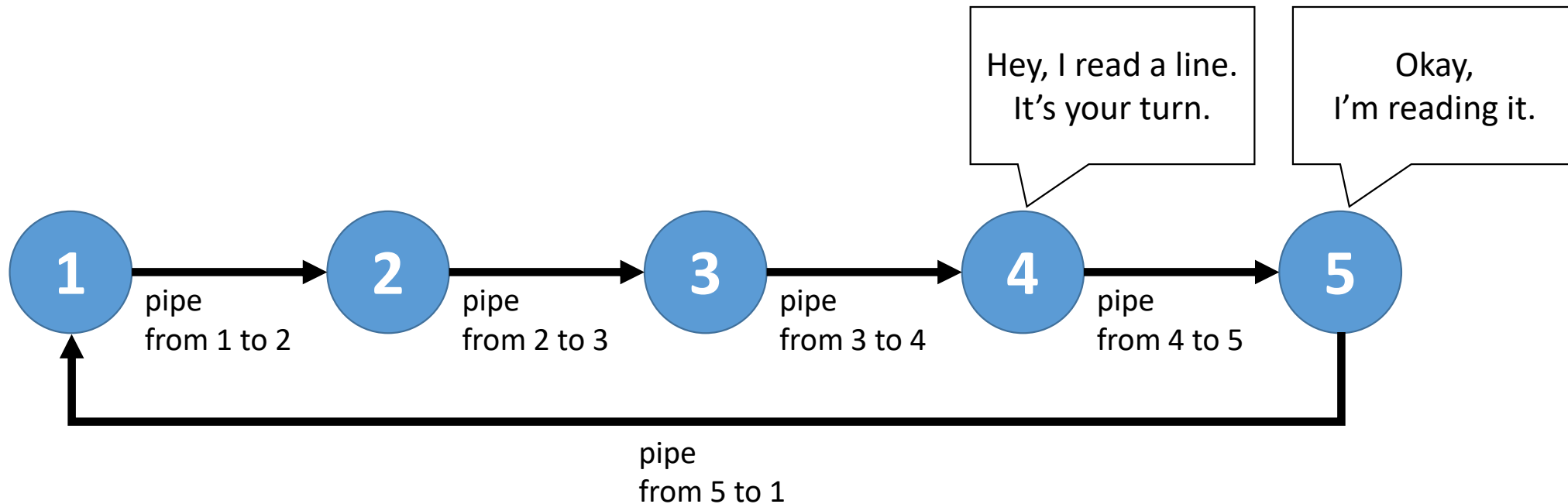4 fourteenth
5 fifteenth
…
…

How can we let the next process read a line?

- PIPE!!

How can we let the next process read a line?

- PIPE!!

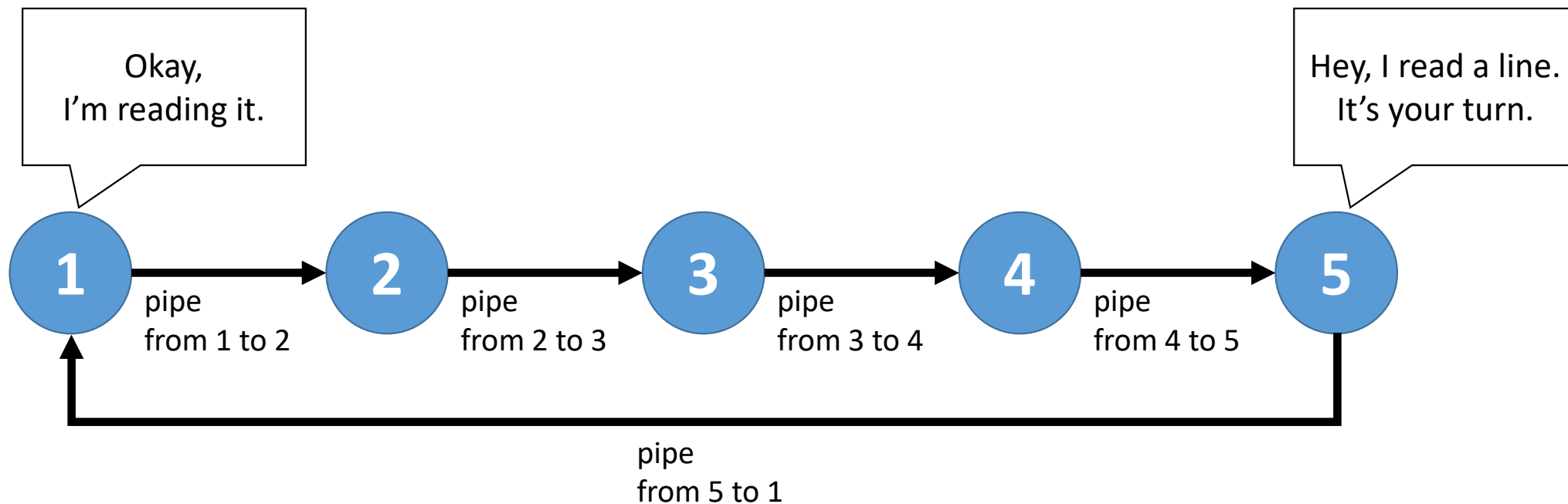# How can we let the next process read a line?

- PIPE!!

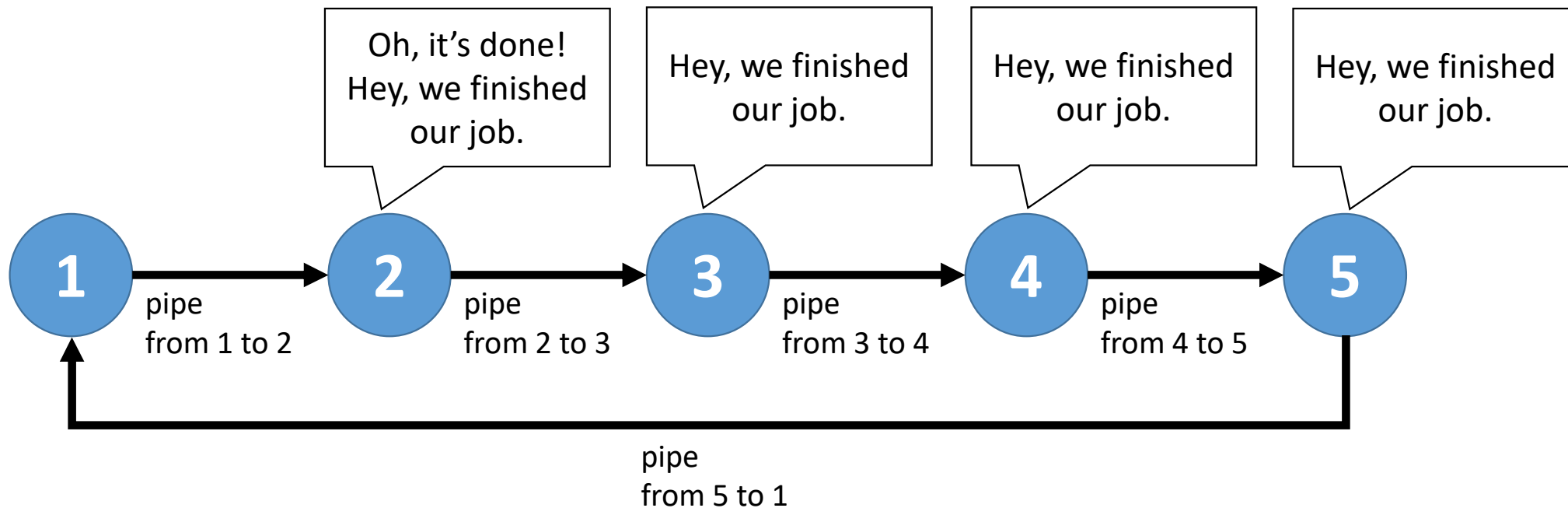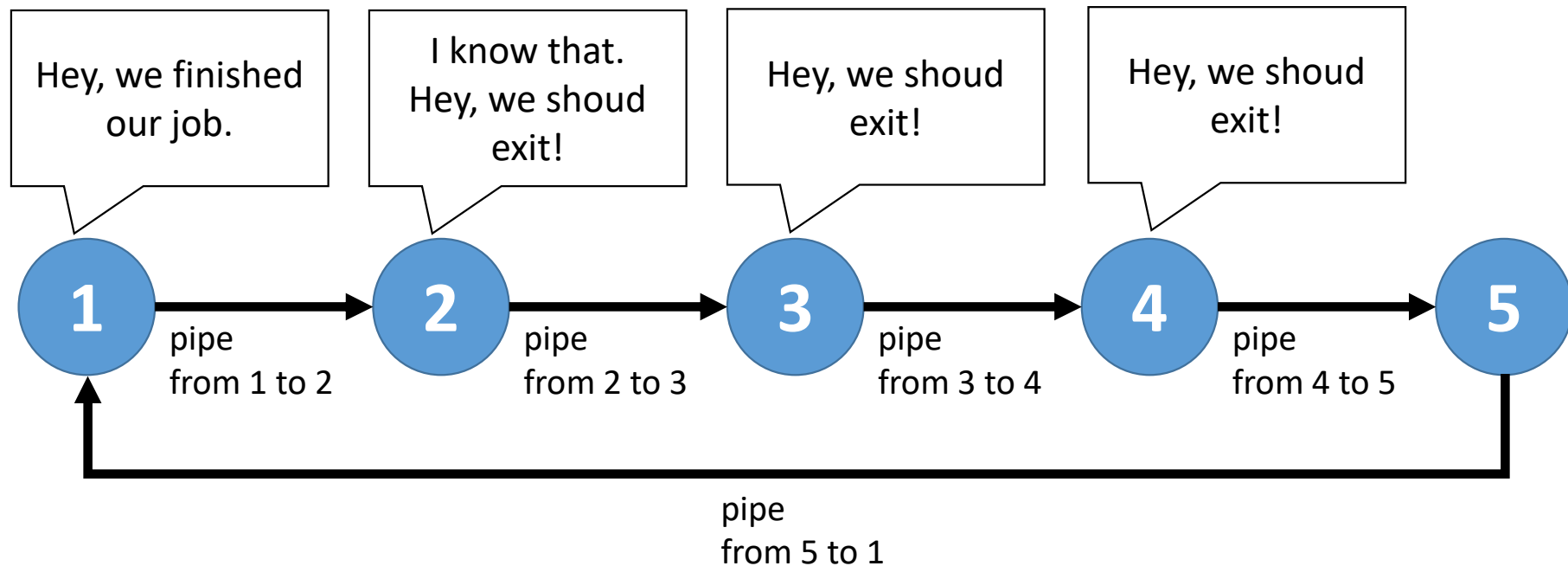# How can we let the next process read a line?

- PIPE!!

How can we let the next process read a line?

- PIPE!!

After reading ALL lines !!
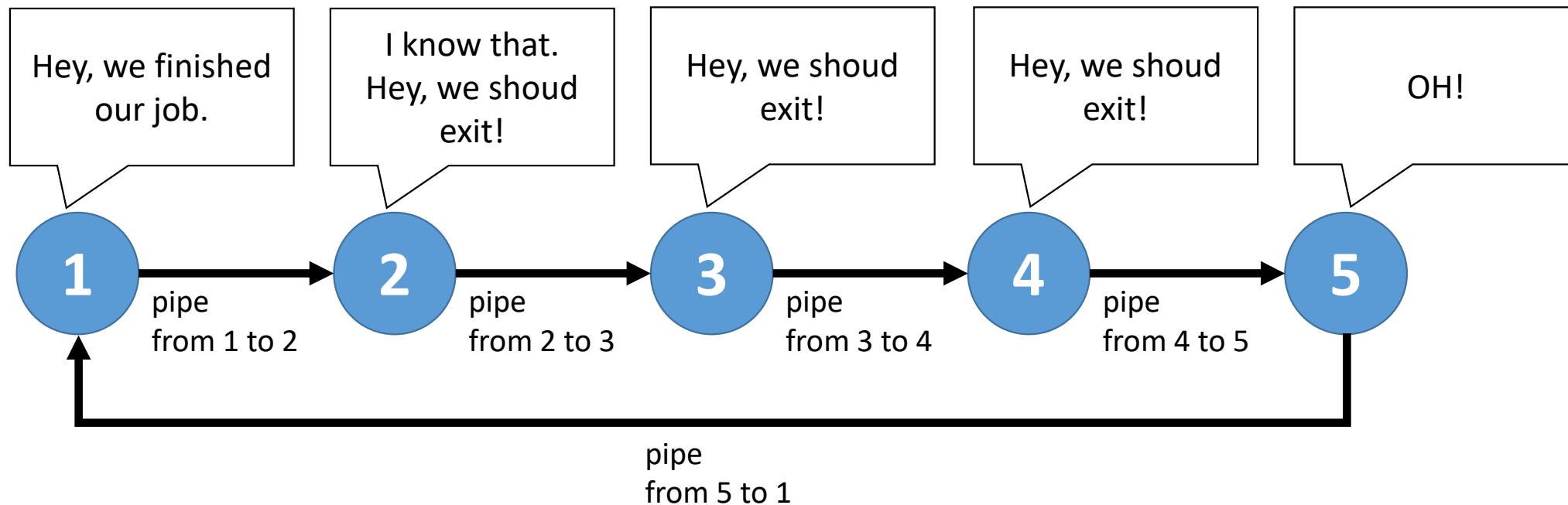
- Exit processes from the last one.

After reading ALL lines !!
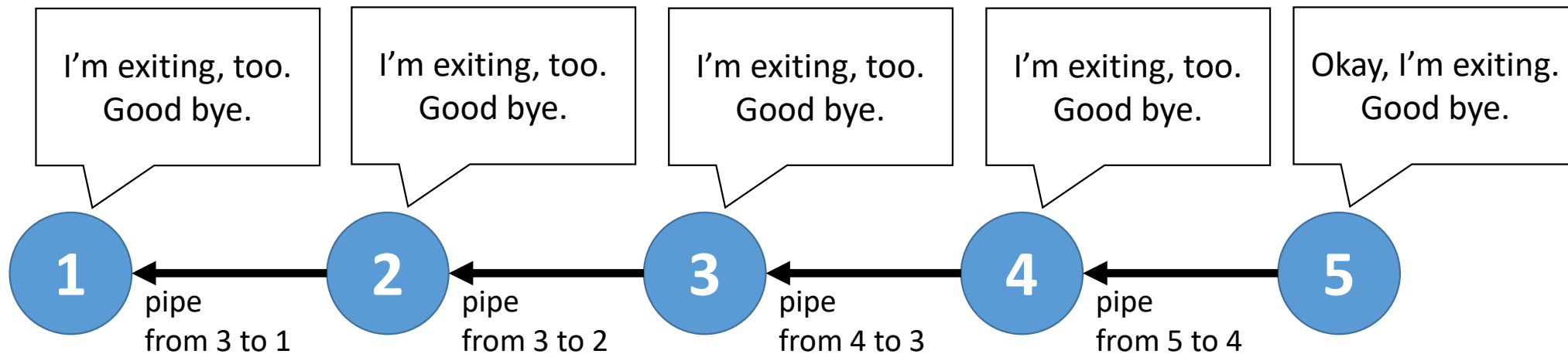
- Exit processes from the last one.

After reading ALL lines !!

- Exit processes from the last one.

After reading ALL lines !!

- Exit processes from the last one.

Why the parent process can't exit first?

Because a child process will be alone in the world.

Output format.

- Your program MUST print out each process's PID and a line of text data

- After reading all lines, a process that read the last line of text data MUST print out its PID with a string "`Read all data`"

- Then, your program MUST exit from the last process, printing out PID with a string "I'm exiting"

Example output of the program

```
1111 first
1112 second
1113 third
1114 fourth
1115 fifth
1111 sixth
1112 seventh
1113 eighth
1114 ninth
1115 tenth
1115 Read all data
1115 I'm exiting…
1114 I'm exiting…
1113 I'm exiting…
1112 I'm exiting…
1111 I'm exiting…
```

Text file

```
first
second
third
fourth
fifth
sixth
seventh
eighth
nineth
tenth
```

File names of Source code and Binary

4. File Names:
```
student-ID#.h <e.g., 20216789.h>
student-ID#.c <e.g., 20216789.h>
Makefile
```
You *MUST* use only the three files for all assignments.

5. Executable File Name: "`student-ID#`"

→ Your Makefile *MUST* generate an executable object file of which file name is your student ID.

→ `<e.g., 20216789>`