

Report

110511010 楊育陞

Part 1

a. test result:

```
• project4 | g++ hw5_1_110511010.cpp -o hw5_1
• project4 | ./hw5_1
Received: 6T7TG6V3A0
Answer: 3667AG0TTV

Received: YC91KITRWI
Answer: 19CIIKRTWY

Received: V837570G2E
Answer: 235778EG0V

Received: TTTXW3GJII
Answer: 3GIIJTTTWX

Received: UKUY4V04Q3
Answer: 0344KQUUVY

Received: Y4167W7QR1
Answer: 114677QRWY

Received: LPF90R2H6E
Answer: 269EFHLOPR

Received: KPCQMWIZDP
Answer: CDIKMPPQWZ

Received: 565Z9VJIQD
Answer: 5569DIJQVZ

Received: Well done!
read_fifo unlinked
write_fifo unlinked
○ project4 | █

• project4 | ./test1.sh
Testcase 1/9 :
Accept!
Testcase 2/9 :
Accept!
Testcase 3/9 :
Accept!
Testcase 4/9 :
Accept!
Testcase 5/9 :
Accept!
Testcase 6/9 :
Accept!
Testcase 7/9 :
Accept!
Testcase 8/9 :
Accept!
Testcase 9/9 :
Accept!
WINNER WINNER CHICKEN DINNER!!
○ project4 | █
```

b. What might happen if your program didn't call `sleep(1)` ? Why? => The program will read the same value from the FIFO, because the program will read the FIFO before the shell finish reading the FIFO and write to it, which is usually slower than the program. And the shell will keep waiting to read.

<pre> ⊗ project4 ./hw5_1_nosleep Received: Z3XA5VDYNB Answer: 35ABDNVXYZ Received: Z3XA5VDYNB ❖ project4 </pre>	<pre> ○ project4 ./test1.sh Testcase 1/6 : Accept! Testcase 2/6 : █ </pre>
---	--

c. What happens when a process writes to a FIFO, but there is no process reading from it? => According to linux manual, the process will be blocked until another process opens the FIFO for reading.

Once you have created a FIFO special file in this way, any process can open it for reading or writing, in the same way as an ordinary file. However, it has to be open at both ends simultaneously before you can proceed to do any input or output operations on it. Opening a FIFO for reading normally blocks until some other process opens the same FIFO for writing, and vice versa. See [fifo\(7\)](#) for nonblocking handling of FIFO special files.

Part 2

a. test result:

<pre> ● project4 g++ hw5_2_110511010.cpp -o hw5_2 ● project4 ./hw5_2 Input Heathcliff's PID: 55679 SIGUSR1 sent successfully ○ project4 </pre>	<pre> ○ project4 gcc test2.c -o test2 ● project4 ./test2 This, might be a game, but it isn't meant to be played. -by SAO Programmer Kayaba Akihiko Heathcliff's PID: 55679 Heathcliff is under attack. You dare use my own spells against me, Kirito? ○ project4 </pre>
---	---

b. How did you defeat Heathcliff ? => Write down the pid of Heathcliff repeatedly for size of 0xC8763 to shared memory, and send SIGUSR1 to Heathcliff.

c. What might happen if you reverse steps 6 and 7, meaning, sending SIGUSR1 before writing the data? => Heathcliff will start to read the shared memory before the data is finished writing, which may cause Heathcliff to read the wrong data and fail to defeat Heathcliff.

<pre> ● project4 ./hw5_2_rev Input Heathcliff's PID: 58576 SIGUSR1 sent successfully ○ project4 </pre>	<pre> ○ project4 ./test2 This, might be a game, but it isn't meant to be played. -by SAO Programmer Kayaba Akihiko Heathcliff's PID: 58576 Heathcliff is under attack. Isn't it quite a dramatic plot development? █ </pre>
---	--