Network Programming Homework 0

Deadline: Wednesday, 2025/3/12 23:55

I.   Description

In this homework, you are given a part of a program and a terminal output of two processes executed simultaneously. You should figure out possible context switch points according to the output.

The left part of page 2 contains a program that periodically reads a number from a file, prints its PID with the number, increases the number by one, and writes the number back to the file. The right part of page 2 shows the output of two processes executed simultaneously (suppose there is only one core). Please analyze the possible situation of every context switch point (marked as (1), (2) ... (9)) and give your answer on page 3.

The **Executing Block** means the section where the process is paused when a context switch happens. You should answer with **"Line x ~ y"** which represents "line x is the last line that we can confirm it's finished and line y must not have been executed". For example, "Line 6 ~ 7" means that "line 6 is the last line that we can confirm it's finished while line 7 must not have been executed", and "Line 10 ~ 6" means that "line 10 is the last line that we can confirm it's finished while line 6 (next iteration) must not have been executed". The **Description** block should contain the reason for your judgment.

You should give the answer based on the output in page 2. DO NOT execute the program. Besides, there may be more than one answer, and you only need to give one reasonable answer with explanation.

II.   Submission
- You should submit your answer to the E3 system.
- Only submit the answer page.
- You should name your file as **[student_id].pdf**, for example, "0856000.pdf".
  ATTENTION! We only accept PDF format.
- Late submissions are not accepted after the deadline.
- DO NOT use handwriting.

Code	Output	

```
 1  int main() {
 2      fstream seq_file("seqno.txt");
 3      int seqno, pid = getpid();
 4      for (int i = 0; i < 20; ++i) {
 5          seq_file.seekg(ios::beg); /* rewind before read */
 6          seq_file >> seqno;
 7          cout << "pid=" << pid << ", seq#=" << seqno << endl;
 8          ++seqno;
 9          seq_file.seekp(ios::beg); /* rewind before write */
10          seq_file << seqno << endl;
11      }
12      seq_file.close();
13  }
```

Output:

```
pid=186, seq#=1        pid=186, seq#=13 ✓
pid=186, seq#=2        pid=186, seq#=14
(1)                    pid=186, seq#=15
pid=187, seq#=3        pid=186, seq#=16
pid=187, seq#=4        pid=186, seq#=17
pid=187, seq#=5        (7)  7~7
(2) 10~6               pid=187, seq#=18
pid=186, seq#=6        pid=187, seq#=19
pid=186, seq#=7        pid=187, seq#=20
pid=186, seq#=8        pid=187, seq#=21
(3) 6~7                (8)  1~1
pid=187, seq#=9        pid=186, seq#=18
pid=187, seq#=10       pid=186, seq#=19
pid=187, seq#=11       pid=186, seq#=20
(4) 1~                 pid=186, seq#=21
pid=186, seq#=9        pid=186, seq#=22
pid=186, seq#=10       pid=186, seq#=23
pid=186, seq#=11       (9)
pid=186, seq#=12       pid=187, seq#=22
(5) 7~7                pid=187, seq#=23
pid=187, seq#=12       pid=187, seq#=24
pid=187, seq#=13       pid=187, seq#=25
pid=187, seq#=14       pid=187, seq#=26
pid=187, seq#=15       pid=187, seq#=27
(6) 10~6
```

Handwritten annotations (blue):
186 W Y ✓ ✓
187 187 wr wr ✓✓ ✗✗
187 W Y ✓ ✗
186 186 w✓ w✓ ✓✓ ✗
187 W Y ✓ ✗

Network Programming Homework 0

Answer

Student ID:

	Executing Blocks		Description
	pid = 186	pid = 187	
(1)	Line 10 ~ 6		由於(1)之後 pid 187 輸出為 3，可知已將 seqno 寫入檔案。 由於(2)之後 pid 186 輸出為 6，可知此時尚未讀取檔案。
(2)			
(3)			
(4)			
(5)			
(6)			
(7)			
(8)			
(9)	Line 7 ~ end		最後一次輸出結束。