**Network Programming Homework 0**

Deadline: Wednesday, 2022/10/06 23:55

# 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 finished and line y must not have been executed”. For example, “Line 6 ~ 7” means that “line 6 is finished while line 7 must not have been executed”, and “Line 10 ~ 6” means that “line 10 is finished while line 6 (next iteration) must not have been executed”. The **Description** block should contain the reason of your judgement.

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.

# 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”.

**ATTENEION! We only accept PDF format.**

* Late submissions are not accepted after the deadline.
* DO NOT use handwriting.

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | | **Output** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | int main() {      fstream seq\_file("seqno.txt");      int seqno, pid = getpid();      for (int i = 0; i < 20; ++i) {          seq\_file.seekg(ios::beg); /\* rewind before read \*/          seq\_file >> seqno;          cout << "pid=" << pid << ", seq#=" << seqno << endl;          ++seqno;          seq\_file.seekp(ios::beg); /\* rewind before write \*/          seq\_file << seqno << endl;      }      seq\_file.close();  } | pid=186, seq#=1  pid=186, seq#=2  **(1)**  pid=187, seq#=3  pid=187, seq#=4  pid=187, seq#=5  **(2)**  pid=186, seq#=6  pid=186, seq#=7  pid=186, seq#=8  **(3)**  pid=187, seq#=9  pid=187, seq#=10  pid=187, seq#=11  **(4)**  pid=186, seq#=9  pid=186, seq#=10  pid=186, seq#=11  pid=186, seq#=12  **(5)**  pid=187, seq#=12  pid=187, seq#=13  pid=187, seq#=14  pid=187, seq#=15  **(6)** | pid=186, seq#=13  pid=186, seq#=14  pid=186, seq#=15  pid=186, seq#=16  pid=186, seq#=17  **(7)**  pid=187, seq#=18  pid=187, seq#=19  pid=187, seq#=20  pid=187, seq#=21  **(8)**  pid=186, seq#=18  pid=186, seq#=19  pid=186, seq#=20  pid=186, seq#=21  pid=186, seq#=22  pid=186, seq#=23  **(9)**  pid=187, seq#=22  pid=187, seq#=23  pid=187, seq#=24  pid=187, seq#=25  pid=187, seq#=26  pid=187, seq#=27 |

**Network Programming Homework 0**

**Answer**

**Student ID: 109550206**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **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)** |  | Line 10 ~ 6 | 由於(2)之後pid 186輸出為6，可知已將seqno寫入檔案。  由於(3)之後pid 187輸出為9，可知此時尚未讀取檔案。 |
| **(3)** | Line 10 ~ 7 |  | 由於(3)之後pid 187輸出為9，可知已將seqno寫入檔案。  由於(4)之後pid 186輸出為9，可知此時已經讀取檔案，但尚未輸出。 |
| **(4)** |  | Line 8 ~ 7 | 由於(4)之前已經輸出了且(4)之後pid 186已經讀取檔案，因此不可考。  由於(5)之後pid 187輸出為12，可知此時已經讀取檔案，但尚未輸出。 |
| **(5)** | Line 8 ~ 7 |  | 由於(5)之前已經輸出了且(5)之後pid 187已經讀取檔案，因此不可考。  由於(6)之後pid 186輸出為13，可知此時已經讀取檔案，但尚未輸出。 |
| **(6)** |  | Line 8 ~ 6 | 由於(6)之前已經輸出了且(6)之後pid 186已經讀取檔案，因此不可考。  由於(7)之後pid 187輸出為18，可知此時尚未讀取檔案。 |
| **(7)** | Line 10 ~ 7 |  | 由於(7)之後pid 187輸出為18，可知已將seqno寫入檔案。  由於(8)之後pid 186輸出為18，可知此時已經讀取檔案，但尚未輸出。 |
| **(8)** |  | Line 8 ~ 7 | 由於(8)之前已經輸出了且(8)之後pid 186已經讀取檔案，因此不可考。  由於(9)之後pid 187輸出為22，可知此時已經讀取檔案，但尚未輸出。 |
| **(9)** | Line 7 ~ end |  | 最後一次輸出結束。 |