ku\_ff.c Design & Implementation

201611218\_이기현

0. 과제 목표

File 을 읽고 쓸 수 있는가

하나의 메인 쓰레드 작업을 멀티 쓰레드 환경으로 분할하여 병행으로 작업할 수 있는가

1. Design & Implementation

큰 그림은 Assign1 과 비슷합니다. 각 프로세스마다 다른 인덱스를 부여 받은 것처럼 쓰레드들은 특정 start 값을 받아서 input.txt 파일의 커서를 lseek 으로 옮긴 후에 한 줄(6Byte)씩, 총 라인 수(lines)만큼 읽습니다. 한 줄씩 읽어 들인 숫자가 조건에 부합하다면 output.txt 에 append 하여 output을 업데이트 해 나갑니다. 각 쓰레드가 센 count값들을 메인 쓰레드에서 받아 total\_count 에 더합니다. 모든 쓰레드가 종료되면 total\_count를 output.txt의 첫번째 줄에 쓰고 프로그램이 종료됩니다. 자세한 구현은 Function Description에 설명이 되어 있습니다.

2. Function Description

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| main | (int argc, char\* argv[]) | 컴파일 후 실행 때 입력 받는 인자들을 파싱하여 예외처리를 하고 전체 프로그램을 실행시킵니다.  전체 순서:  init() 파일 초기화  thread 개수만큼 쓰레드 생성  thread 개수만큼 쓰레드 조인  output파일 write  종료 |
| Return 0; |

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| init | (void) | input\_name 의 파일을 O\_RDONLY 모드로 open하고, output\_name의 파일을 O\_WRONLY 모드로 open합니다.  read open에 성공하였으면 첫 번째 라인을 6바이트만큼 읽고 파싱하여 input\_count에 저장해 둡니다.  write open에 성공하였으면 첫째줄에 공백을 포함한 0을 써서 output 파일을 초기화해줍니다.  저장해두었던 input\_count를 리턴하여 메인에서 쓸 수 있도록 합니다. |
| Return input\_count; |

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| read\_and\_write | (void\* data) | 메인에서 적절히 부여 받은 start 값과 lines 값을 인자로 전달받아 실행하는 쓰레드의 메인 함수입니다. 쓰레드 메인의 형식에 맞게 인자, 리턴값 모두 (void\*) 입니다.  전달할 인자가 (start, lines) 두 개이기 때문에 Arg 라는 struct를 선언하여 인자로 Arg를 전달합니다.  start는 각 쓰레드가 input.txt파일을 읽을 때 시작하는 커서 위치입니다. lseek 함수를 이용하여 read open된 파일의 커서를 옮기고 lines 수만큼 라인을 읽으면서 조건 체크 후 count증가, output에 append합니다.  예를 들어, 처음으로 생성된 쓰레드가 start= 1 과 lines = 4를 받았다면 input.txt 의 start\*6으로 커서를 옮기고 4줄을 읽어 나갑니다.  count를 다 셌으면 형변환 후 리턴하여 메인에서 total\_count에 더할 수 있도록 합니다. |
| Return (void\*)count; |

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| write\_total\_count | (int total\_count) | 각 쓰레드가 읽어들인 count 값들을 계속 더하여 얻어진 total\_count를 인자로 받아서 output.txt의 첫 째 줄에 씁니다. 첫 째 줄이기 때문에 lseek 0으로 커서를 옮겨줍니다.  total\_count 에 쓰레드 하나가 읽은 count를 더하는 과정에서 다른 쓰레드에 의해 선점 당하여 race condition 이 일어날 수 있기 때문에 total\_count 업데이트 구문 사이에 mutex lock를 걸어 동기화를 해줍니다. |
| Return void; |

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| itoa | (int num, char\* line) | 조건에 맞는 숫자(int)를 output.txt에 write 할 때 char\* 로 형변환을 하여 넣어야 하는데 그때 필요한 함수입니다.  문자를 int로 형변환해주는 atoi는 잘 작동하는데 반대인 itoa는 정규 함수로 인식을 못해서 직접 만들었습니다.  5바이트에 맞게 칸을 만들고 일의 자리부터 num의 자리수까지 하나씩 char를 대입하는 방식입니다. |
| Return void; |

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Arguments** | **Description** |
| digit | (int n) | n의 자리수를 반환하는 함수입니다.  itoa 함수에서 num의 자리수를 구할 때 사용됩니다. |
| Return dig |