****

컴퓨터 정보 공학과 3학년

12131579 이진아

**오퍼레이팅 시스템**

**1차 과제**

담당 교수: 송민석 교수님

**1. 개요**

구현 목표 : 리눅스 에서 mutithread를 생성 하여 숫자 배열을 정리하는 프로그램을 짜며 thread함수들을 익히고 multithread processing의 개념을 이해한다.

**2. 용어 정리**

**1) multithread**

**스레드**(thread)는 어떠한 프로그램 내에서, 특히 프로세스 내에서 실행되는 흐름의 단위를 말한다. 일반적으로 한 프로그램은 하나의 스레드를 가지고 있지만, 프로그램 환경에 따라 둘 이상의 스레드를 동시에 실행할 수 있다. 이러한 실행 방식을 멀티스레드(multithread)라고 한다.

**2) multithread의 장점**

**응답성** : 대화형 프로그램을 멀티스레드화하면, 프로그램의 일부분(스레드)이 중단되거나 긴 작업을 수행하더라도 프로그램의 수행이 계속되어, 사용자에 대한 응답성이 증가된다. 예를 들어, 멀티스레드가 적용된 웹 브라우저 프로그램에서 하나의 스레드가 이미지 파일을 로드하고 있는 동안, 다른 스레드에 사용자와의 상호작용이 가능하다.

**자원 공유** : 스레드는 자동적으로 그들이 속한 프로세스의 자원들과 메모리를 공유한다. 코드 공유의 이점은, 한 응용 프로그램이 같은 주소 공간 내에 여러 개의 다른 활동성 스레드를 가질 수 있다는 점이다.

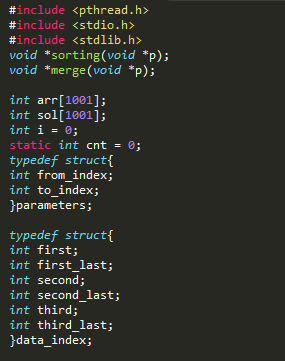
**경제성** : 프로세스 생성에 메모리와 자원을 할당하는 것은 비용이 많이 든다. 스레드는 자신이 속한 프로세스의 자원들을 공유하기 때문에, 스레드를 생성하고 문맥교환을 하는 편이 보다 경제적이다.

**멀티프로세서 활용** : 멀티프로세서 구조에서는 각각의 스레드가 다른 프로세서에서 병렬로 수행될 수 있다. 단일 스레드 프로세스는 CPU가 많아도 CPU 한개에서만 실행된다. 즉, 다중 스레드화를 하면 다중 CPU에서 병렬성이 증가된다.

출처: 위키백과

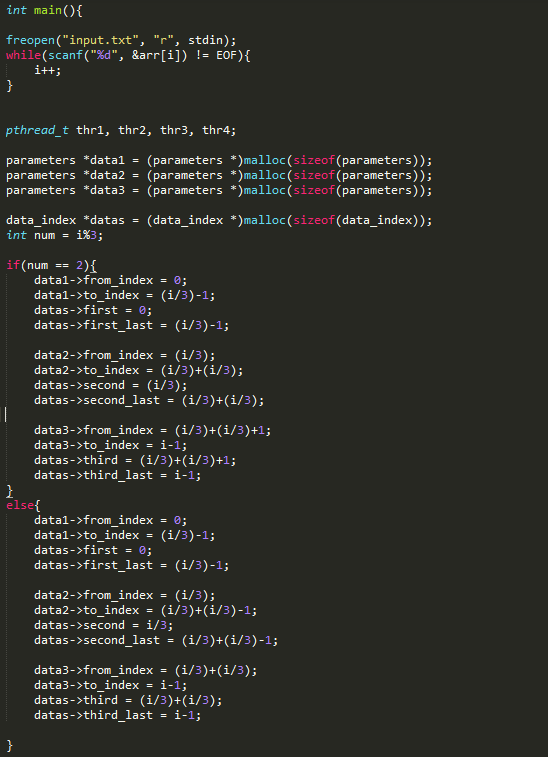
**3.코드 설명**

**1)구조체 및 변수 생성**



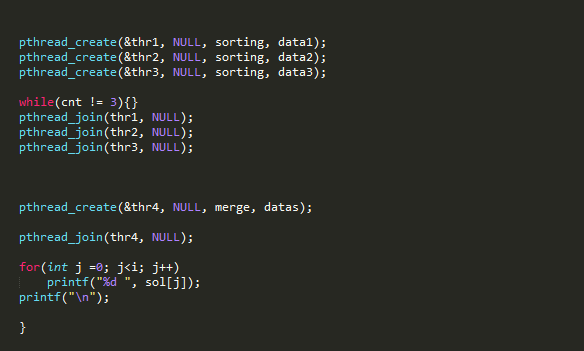
arr배열은 값을 입력 받는 배열이고 sol은 마지막으로 답을 도출 할 때 사용하는 배열이다. i변수는 전체 들어오는 수의 개수를 세기 위해 사용한 변수이다. Cnt는 spin lock을 사용할 시 모든 thread의 sorting이 끝났는지 확인 하기 위해 사용하는 변수이다. Parameters 구조체는 sorting해야 하는 배열의 처음과 끝부분을 저장하는 struct이고 data\_index는 merge시 세개의 thread로 나눠지는 배열의 처음과 끝부분을 저장 하는 struct이다.

**2)입력과 초기화**



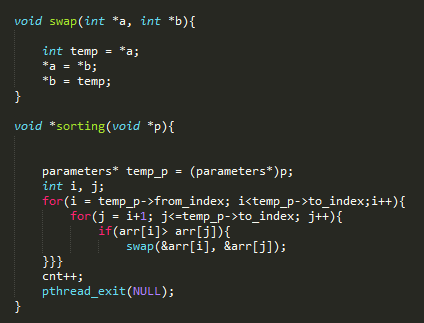
While 조건문 안에 입력값을 넣어서 EOF가 들어올 때 까지 입력을 받도록 처리하였다. 그후 thread를 4개 선언 한 수 parameter struct를 3개를 동적할당 해 주었고 data\_index struct를 하나 동적 할당 해 주었다. 만약 전체 받은 i의 값이 3으로 나누었을 때 2라면 3개로 나누는 부분에서 두부분이 나머지보다 하나가 많아야 하므로 따로 처리를 해주었다. 동적 할당 받은 parameter 구조체에 sorting을 해줄 처음과 끝부분을 넣어 주었고 data\_index 구조체에는 처음, 두번째, 세번째 나누어 sorting을 하는 처음과 끝부분을 넣어 주었다.

**3)thread생성 및 spin lock**



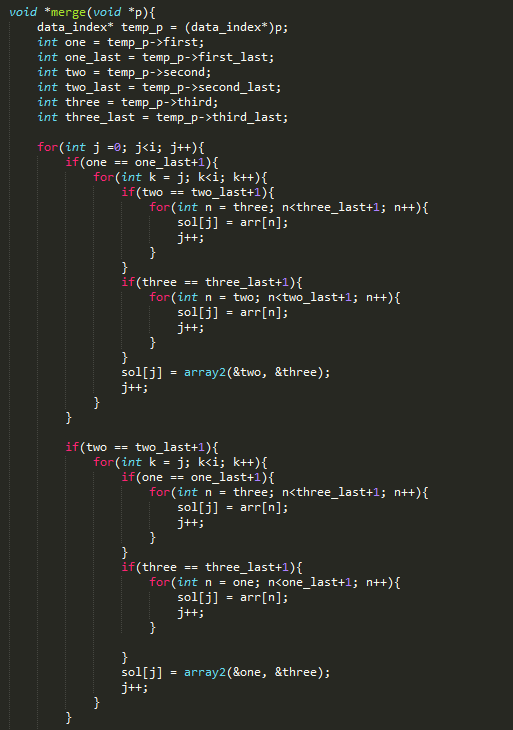
Pthread\_create함수를 통해 thread를 생성 하여 parameter 구조체 형식을 가진 파라미터를 가지고 sorting함수를 실행하도록 해주었다. 세개의 thread가 다 끝났는지 확인 하기 위해 중간에 while문을 넣어주었다. 세개의 thread가 다 수행이 되면 pthread\_join을 통해 끝난 thread를 받아왔다. 각각의 thread의 sorting을 마치고 난 후 merge를 해주기 위한 thread를 생성하여 merge함수를 수행하였고 마지막으로 정렬된 값들을 출력해 주었다.

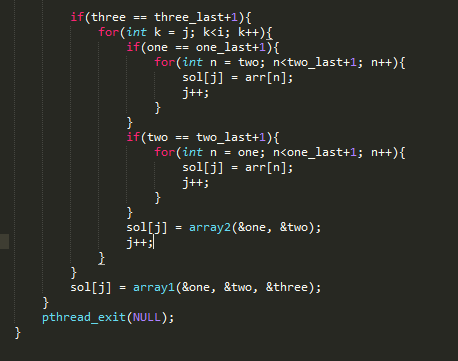
**4)sorting함수**



Sorting을 하기 위해 넘겨준 parameter 구조체를 받는다. 구조체의 처음 부분부터 끝부분 까지 bubble sort를 이용하여 정렬을 해준다. 이때 swap함수를 사용하였는데 arr[i]와 arr[j]의 값을 바꿔주기 위해 사용하였다. 수행이 다 끝난 후 cnt를 하나 증가 시켜주었다. 이는 sort가 다 끝났음을 의미하므로 pthread\_exit를 사용해 thread를 종료시켰다. Cnt는 spin lock에서 thread세개가 다 끝났음을 확인할 때 사용되는 변수이다.

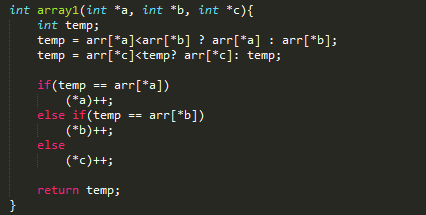
**5)merge함수**

****

****

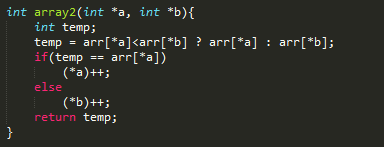
Data\_index 타입의 구조체를 받아 merge함수를 수행한다. 세 부분으로 나눈 처음과 끝부분을 변수에 받고 merge를 시작한다. 만약 세 부분으로 나눈 곳 모두 끝부분 까지 가지 않았으면 array1함수를 수행하여 가장 최소 값을 찾는다. 찾은 최소 값은 결과 값을 저장하는 sol배열에 넣어 준다. 만약 세 부분 중 하나의 부분이 끝까지 갔다면 if문 안쪽으로 들어가게 된다. 세 부분 중 하나의 부분만 끝이 나고 나머지 두 부분이 끝나지 않았으면 array2를 수행하여 최소값을 찾아 sol배열에 넣어준다. 마지막으로 두 부분이 끝까지 확인 하였다면 남은 하나의 부분은 남은 부분을 다 sol배열에 차례대로 넣어준다.

**6)array1함수**

****

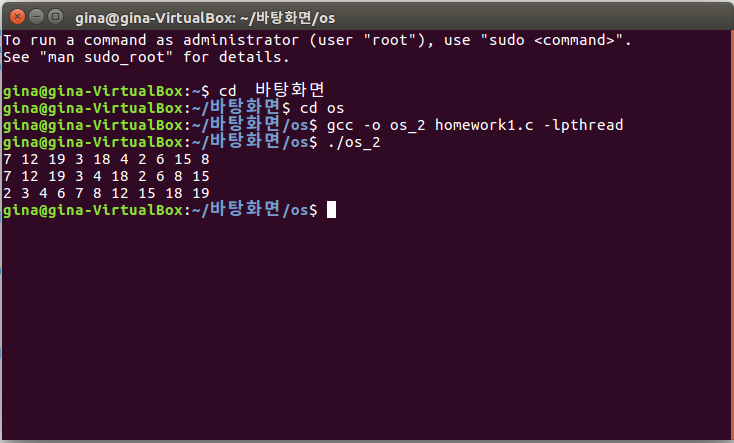
세 부분의 최소 값들을 비교 하여 가장 작은 최소값을 구하는 함수이다. 세 부분의 최소 값이 넘어오면 가장 작은 값을 temp에 저장한다. 그 후 temp와 같은 값을 갖는 부분 즉 세 부분의 최소값 중 가장 작은 값을 가진 곳의 index 값을 하나 증가시킨다.

**7)array2 함수**



세 부분 중 하나의 부분이 끝까지 확인을 하여 두 부분만 확인 하면 될 때 사용하는 함수 이다. 두 부분의 최소 값이 넘어오면 그 중 가장 작은 값을 temp에 저장한다. 그 후 가장 작은 값인 temp와 값이 같은 곳의 index를 하나 증가시킨다.

**4. 실행결과**

****

첫 줄이 입력 들어온 값이고 두번째 줄이 sorting함수 까지 진행 했을 때의 결과이다. 마지막 세번째 줄은 merge함수 까지 실행 하여 전체적으로 정렬을 한 값이다.