**Pintos Project 3 : Thread**

**(설계 프로젝트 수행 결과)**

과목 명 :운영체제

담당 교수 :김영재

조 / 조원 : 10조 / 황순, 허준형

개발 기간 :2018.11.25 ~ 2018.12.03

**프로젝트 제목 : pintos project 3 Thread**

**제출일 : :2018.12.03**

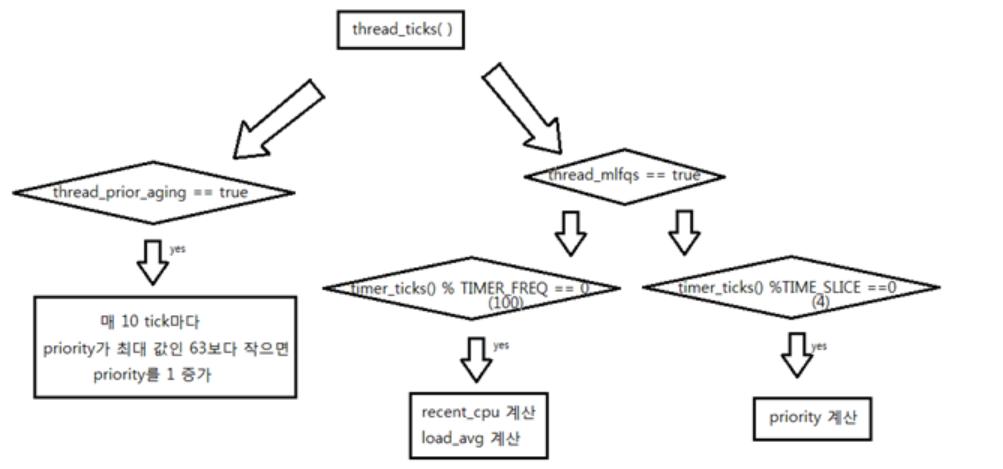
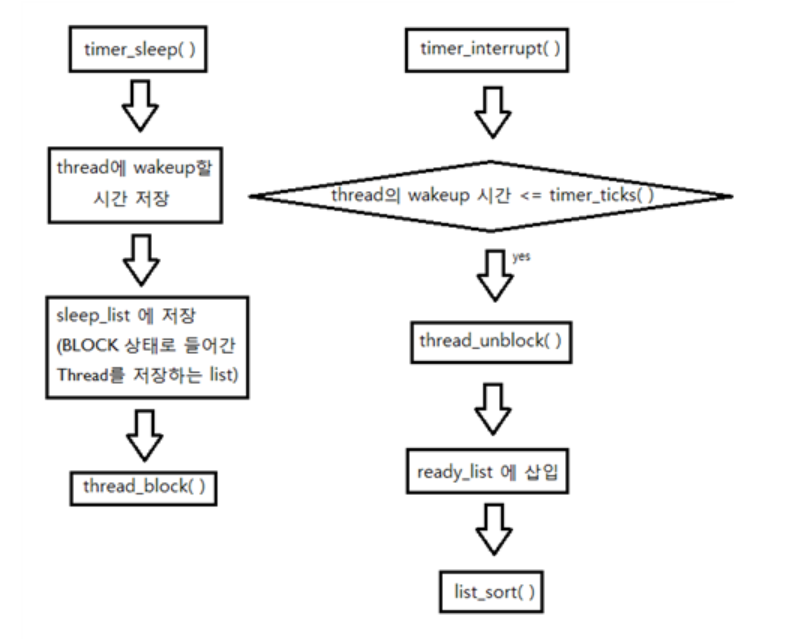
**참여 조원 : 황순 / 허준형**

1. **개발 목표**

* **현재 핀토스는 thread 스케줄링이 최적화 되어있지 않고, busy-waiting 상태로 스케줄링이 구현되어있다. Thread를 block 상태로 만들고 깨울 시간이 되면 ready queue에 삽입하는 방식으로 구현한다. 이를 세마포어와 aging 기법등을 통해 우선순위를 구현하는 최적의 스케줄링을 구현하고 나아가 mlfq 기법을 구현하여 thread 스케줄링을 완성하는 것이 목표이다.**

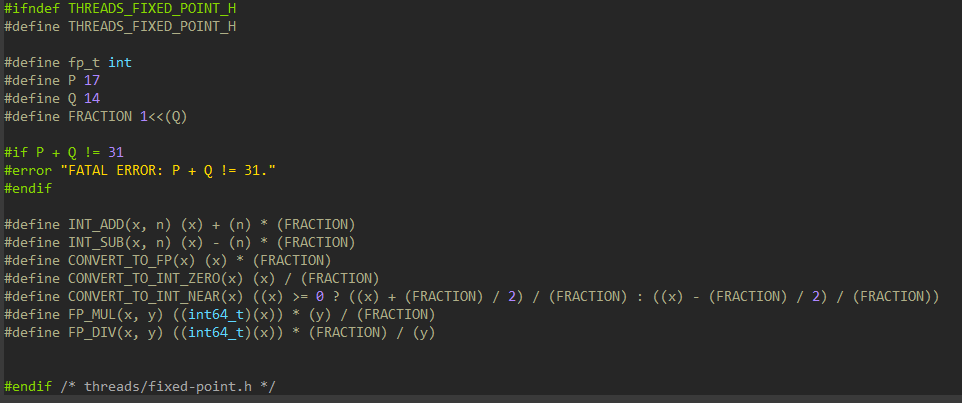
1. **개발 범위 및 내용**
   1. **개발 범위**

* **1. Alarm clock 의 timer\_sleep() 부분을 비지 웨이팅이 아닌 다른 방법으로 바꾼다. (Wake up 과 레디 큐 삽입을 이용한다.)**
* **2. 핀토스는 정수연산만 지원되므로, fixed-point 연산을 통해 실수연산을 하게 한다. 우선순위를 적용시킬 때 실수연산이 필수적이다.**
* **3.. 우선순위를 적용하여 스케줄링을 구현한다.**
* **기존의 FIFO 스케줄링이 아닌 BSD 스케줄러를 구현한다.**
* **4. 우선순위를 적용하기 위한 여러 가지들을 구현한다.**
* **5. 성능을 체크한다.**
  1. **개발 내용**

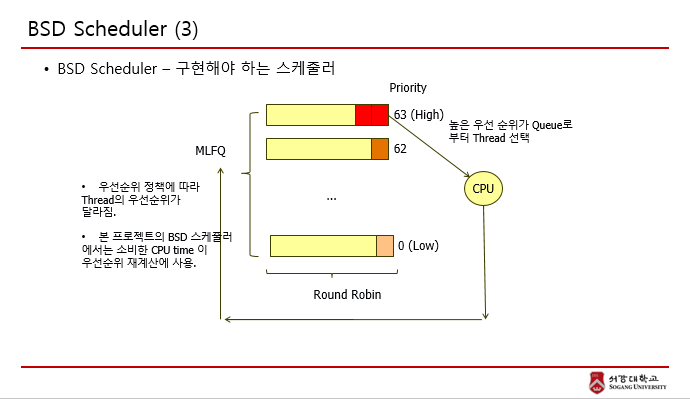


**threads/fixed-point.h**

* **실수연산을 위한 fixed-point 구현**

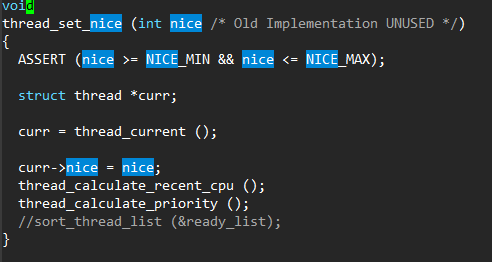


* **구현해야하는 bsd scheduler.**



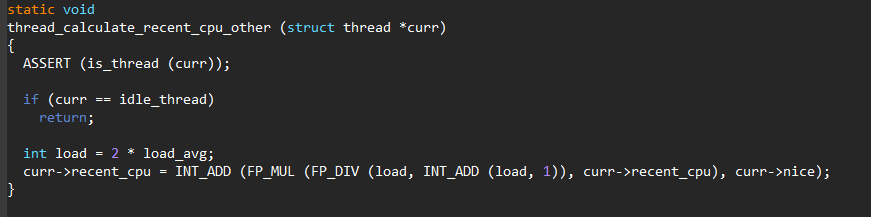
**BSD scheduler 를 위한 값들**

* + 1. **Niceness 구현**

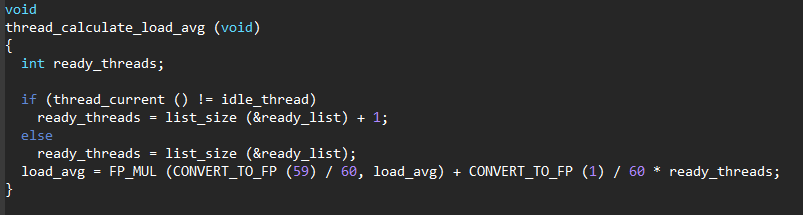


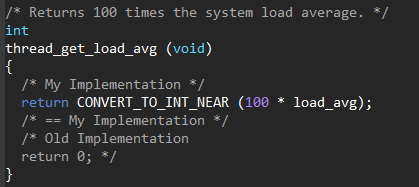
* + 1. **Recent-cpu 구현**

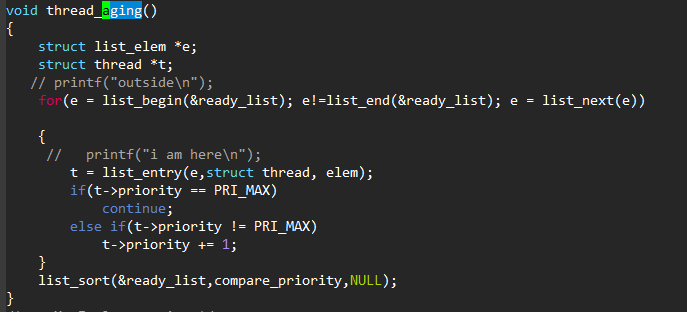
**threads/thread.c**



* + 1. **Load\_avg 구현**





* + 1. **Aging 구현**

**이후 timer.c 와 synch.c 에서 스케줄링을 구현한다.**

1. **추진 일정 및 개발 방법**
   1. **추진 일정**
      1. **Fixed-point 를 구현하여 실수연산을 되게 한다.**
      2. **BSD 스케줄러에서 필요한 값들을 thread.h 에 설정한다.**
      3. **Thread.h 에 필요한 스케줄링에 필요한 값들을 계산해주는 함수를 만들고 이를 적용한다.**
      4. **성능을 테스트한다.**

**구현해야할 스케줄링 구현은 다음과 같다.**

* + - 1. **Alarm Clock.  
         - 시간체크를 한후 시간이 아직안되면 쓰레드를 블록상태로 만든후, 깨울 시간이 되었으면 레디큐에 삽입하는 방식으로 구현.**
      2. **Priority Scheduling  
         - 우선순위 기준을 nice, recent\_cpu 값등을 통해 레디큐에 삽입시 우선순위 대로 정렬하게 한다. List\_push\_back, list\_insert\_ordered를 통해 정렬된 상태로 큐에 삽입되게 한다.**
      3. **Aging  
         - starvation 현상을 막기위해 오래도록 큐에 남아있으면 우선순위를 계속 높여서 결국엔 실행되게 만든다**
      4. **BSD 스케줄러를 구현한다.  
         - pintos manual을 참조하여 구현한다.**
  1. **개발 방법**

**평소 매뉴얼을 각자 읽으면서 개념을 익힌후 평일 날마다 모여서 코딩을 하였다.**

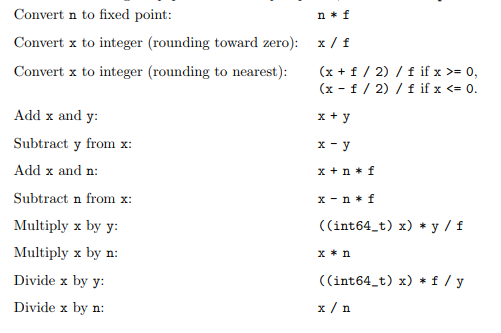
**이번에도 역시 전통적인 agile 방식을 통하여 구현하였다.**

* 1. **연구원 역할 분담**

|  |  |
| --- | --- |
| * **황순** | **BSD 에서 nice, load avg,등 구현, mlfq 구현, 보고서작성** |
| * **허준형** | **Fixed-point 구현, mlfq 구현, 보고서작성** |

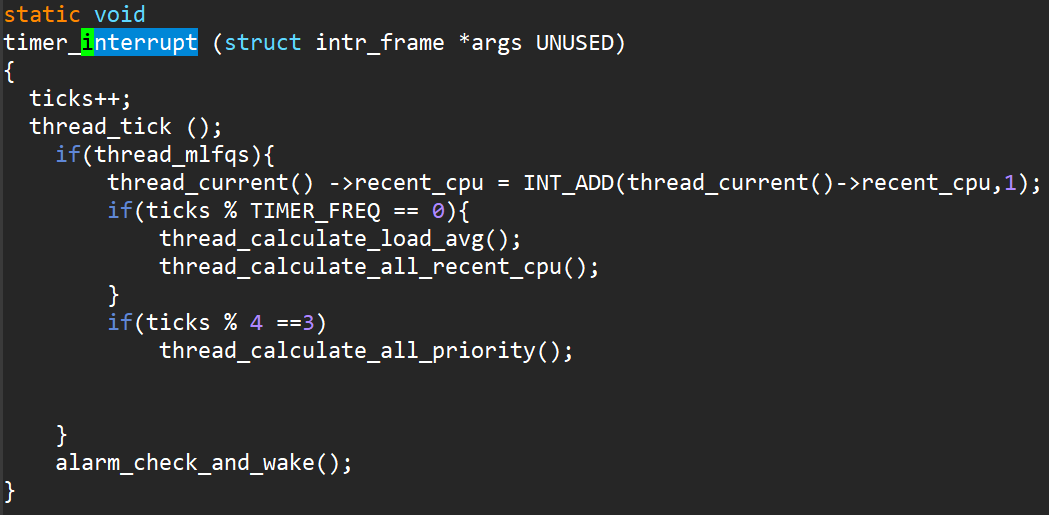
1. **연구 결과**
   1. **합성 내용**

**처음 한 것은 fixed-point.h 구현이었다. Pintos manual 에 따라 fixed-point를 구현하였다.**



**원래 부동소수점이랑은 조금 다르다는 것을 배우게 되었다. 이를 이용하여 쓰레드의 값들과 우선순위를 계산할 수 있게 됐다.**

**그 이후 timer\_interrupt 를 구현하였다. 여기서는 스케줄링에 필요한 값들을 계산해주고 나서, 재울 쓰레드를 일정한 timer\_tick마다 기준을 잡아서 alarm을 해제해주는 alarm\_check() 라는 함수를 구성하여 구현하였다. 코드는 다음과 같다.**

* 

**이후 나머지 스케줄링에 필요한 변수 계산은 위의 코드로 대신하겠다.**

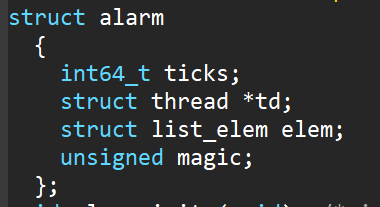
**프로그램의 순서도는 다음과 같다. 여러 쓰레드가 생선되면, 그것은 multi level feedback queue 에 우선순위를 갖춰서 저장이 된다. ( 64 level의 우선순위, 통칭 bsd 스케줄러)**

**이 때 매 timer\_interrupt가 걸릴 때 마다. 각 thread의 스케줄링을 위한 priority가 계산된다. 이 씨피유의 사용시간 추정값( recent\_cpu), nice, 그리고 aging이 있다. 씨피유를 가장 많이 쓰는 것 같은 쓰레드는 우선순위가 낮아진다는걸 알 수 있다. 여기에 load-avg(레디상태 쓰레드 개수의 평균) 을 적용하여, 다양한 우선순위를 고려한다.**

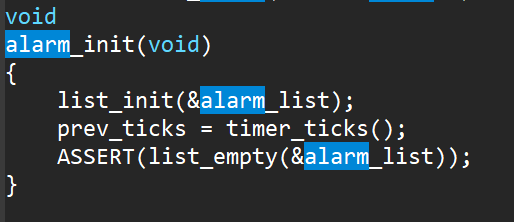
**그러나 이 우선순위는 잘못하면, starvation을 야기할 수 있다. 우선순위가 낮은쓰레드가 영원히 실행 안될 수도 있는 경우가 있다. 이것을 방지하기 위해 흐른 시간에 비례해서 priority를 높여주는 aging기법을 적용한다.**

**이제 이것들을 완성하면, 우리가 원하는 프로젝트가 완성이다.**

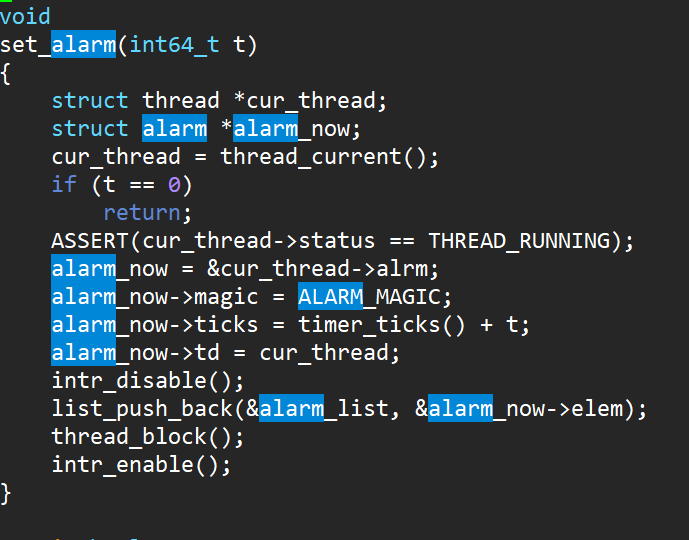
* 1. **제작 내용**



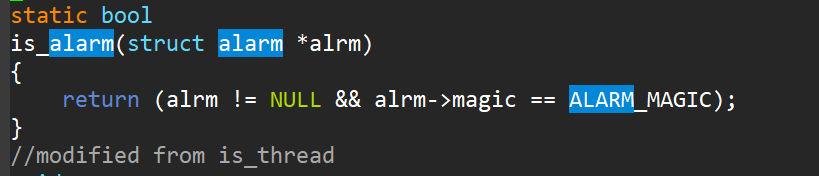
Alarm 관련 함수 해결을 위해 새로운 구조체 alarm을 선언하였다.



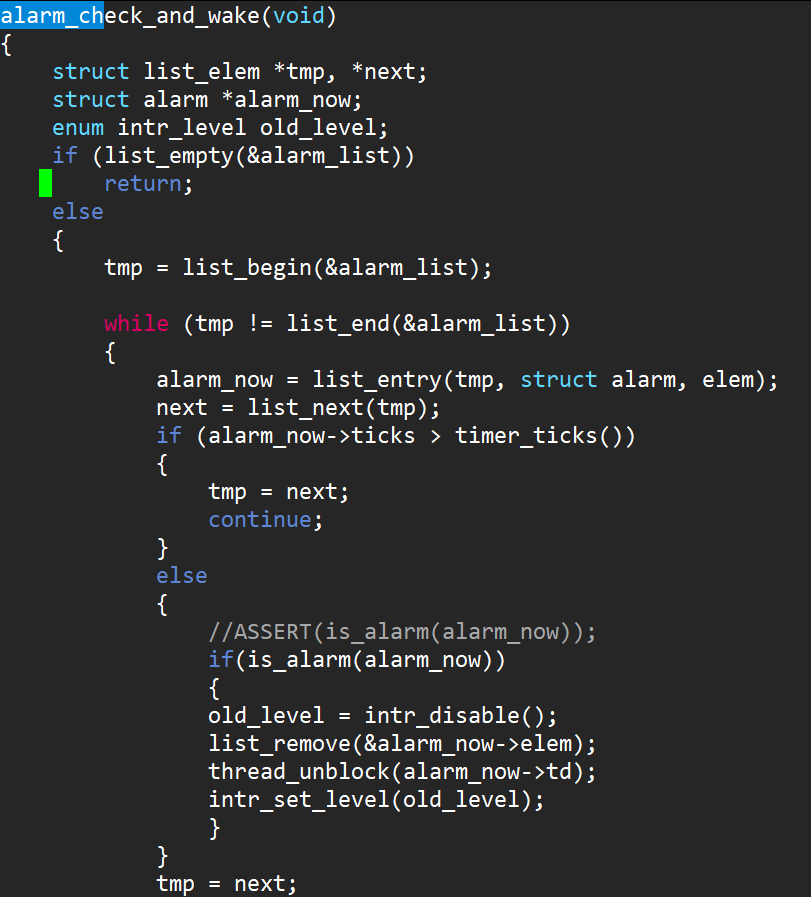
Alarm list를 init해주는 함수이다.



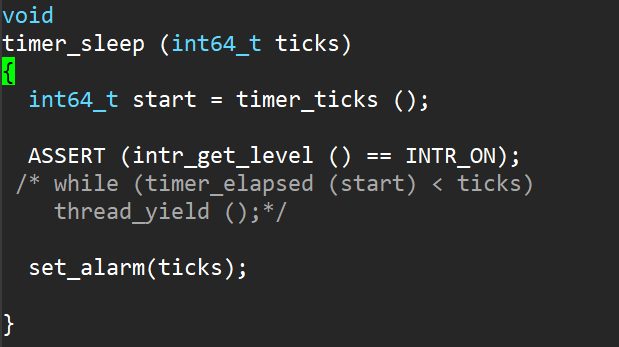
현재의 쓰레드에 alarm을 설정해주는 함수이다.



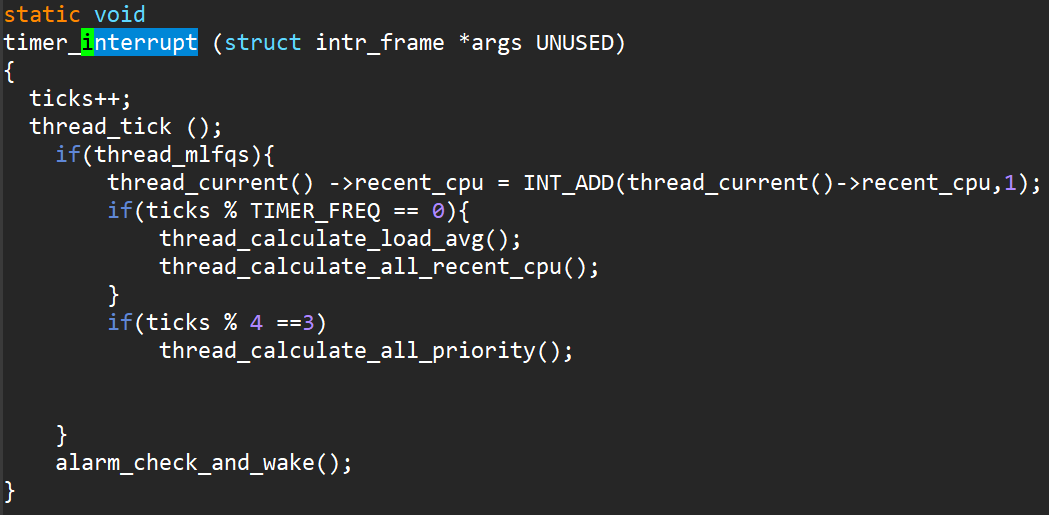
alarm인지 아닌지 체크해주는 함수이다.



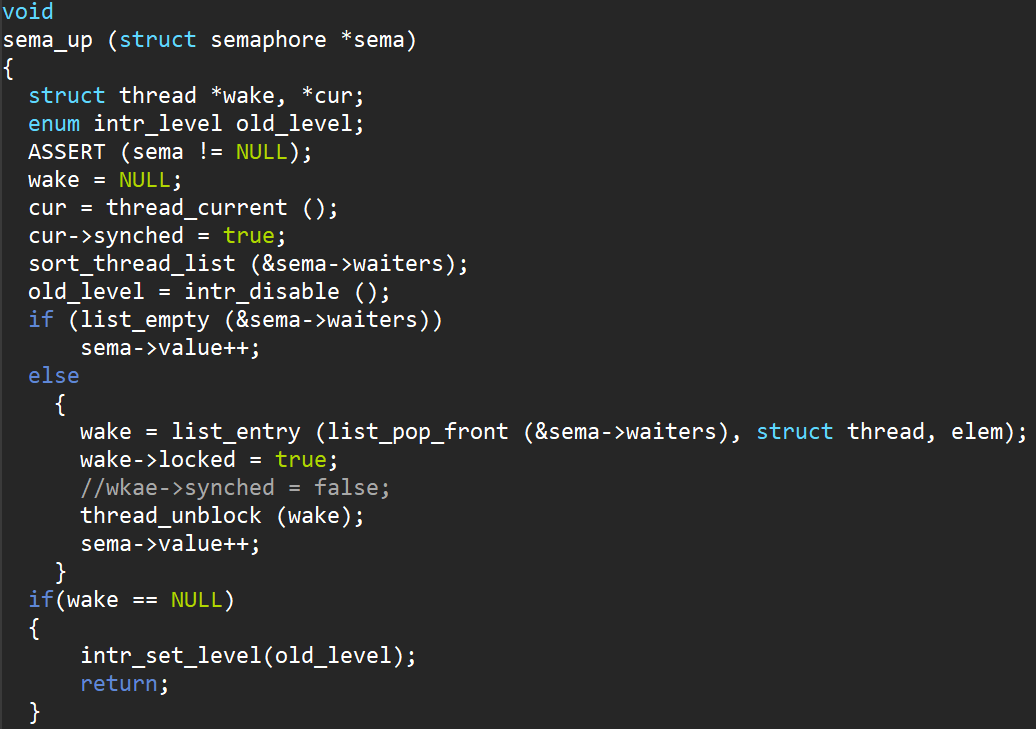
Alarm list를 체크하여 깨울 시간이 된 alarm을 깨워주는 함수이다.

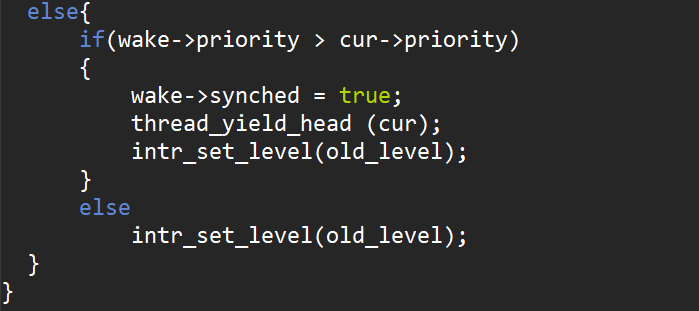


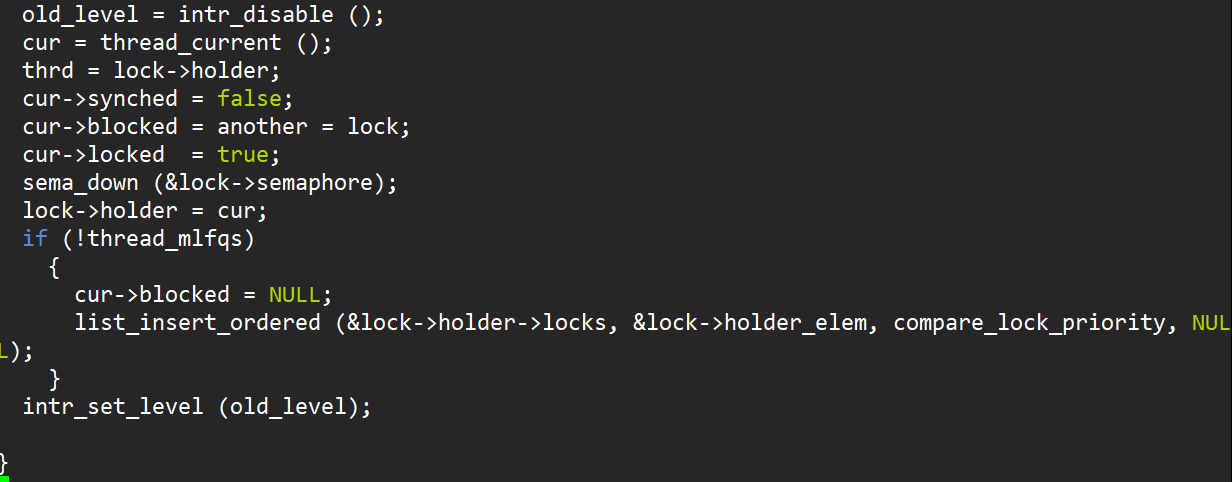
새롭게 수정된 timer\_sleep함수이다. 여기서 set\_alarm을 통해 알람을 설정하면,



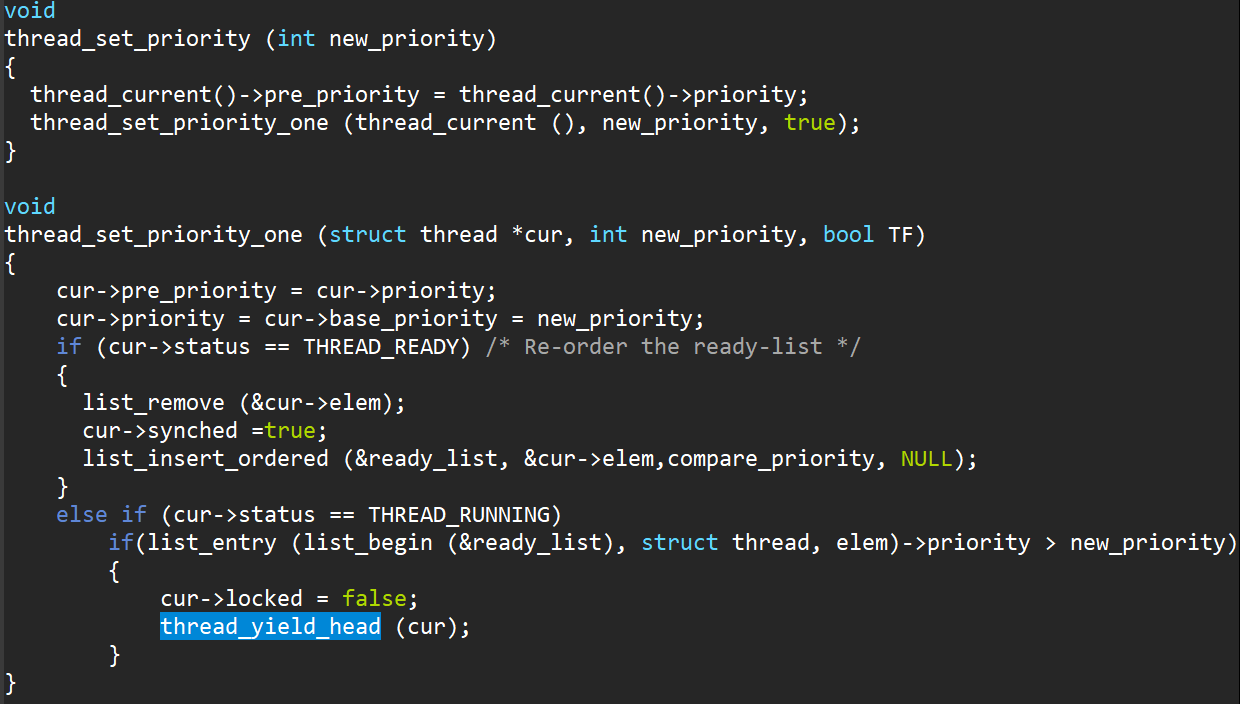
Timer\_interrupt에서 alarm\_check를 하고 깨울시간이 된 alarm을 깨운다.



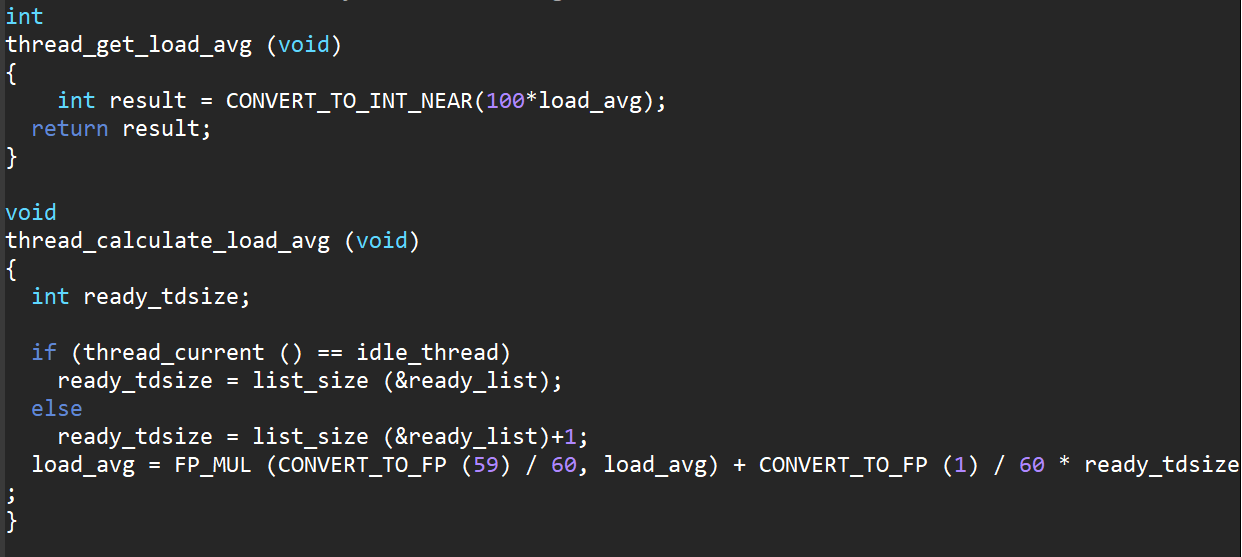




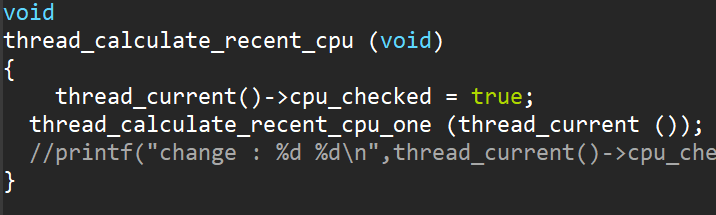
Priority\_sema 구현을 위하여 sema\_up함수를 수정하였다. Thread\_yield\_head이용을 통해 priority를 비교하여 insert해준다.

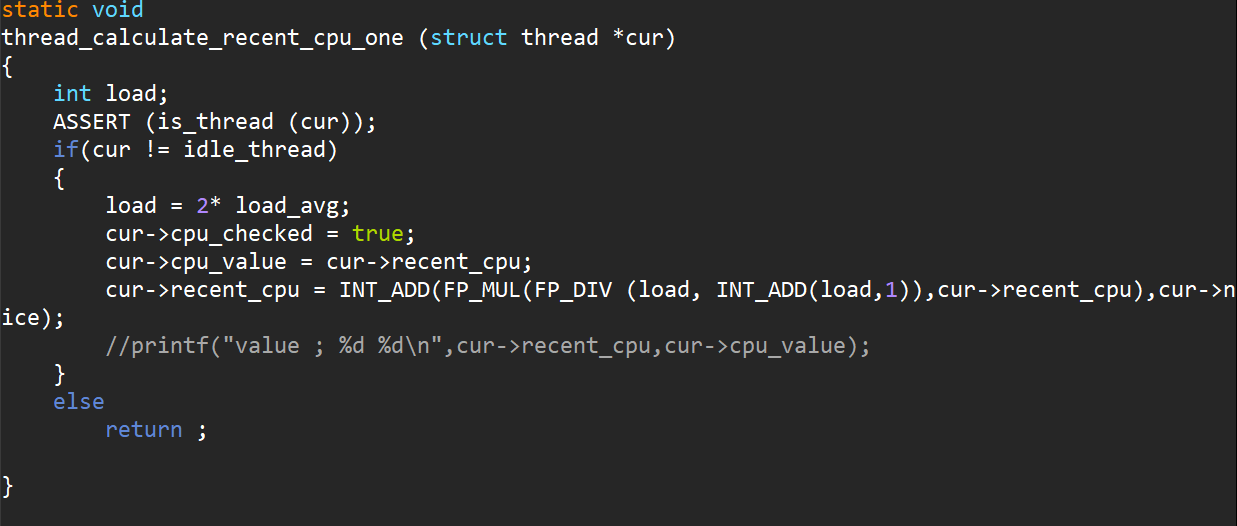


Priority 구현을 위하여 다음과 같은 함수들을 선언하였다. 현재의 쓰레드의 status를 확인하고 running이면 thread\_yield\_head를 통해 idle인지 아닌지 확인하여 insert해주고 ready이면 ready\_list에 priority에 따라 정렬하여 넣어주고 list에서 remove한다.

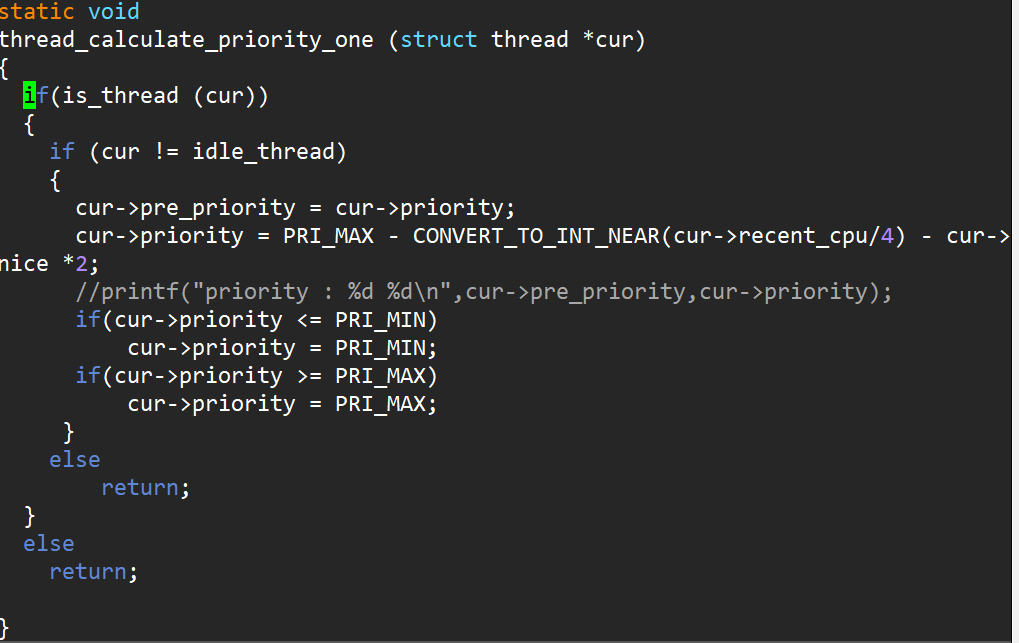


쓰레드의 load\_avg를 계산하기 위한 함수이다. FP\_MUL, CONVERT\_TO\_FP 는 fixed-point.h에 정의되어있다.

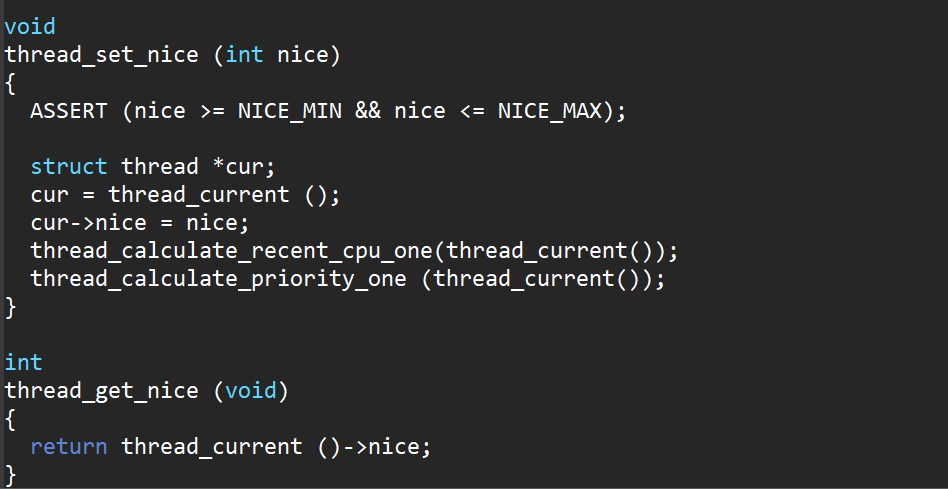




최근에 사용된 recent cpu의 load값을 fixed-point.h에 저장된 매크로를 이용하여 계산해 주는 함수이다.



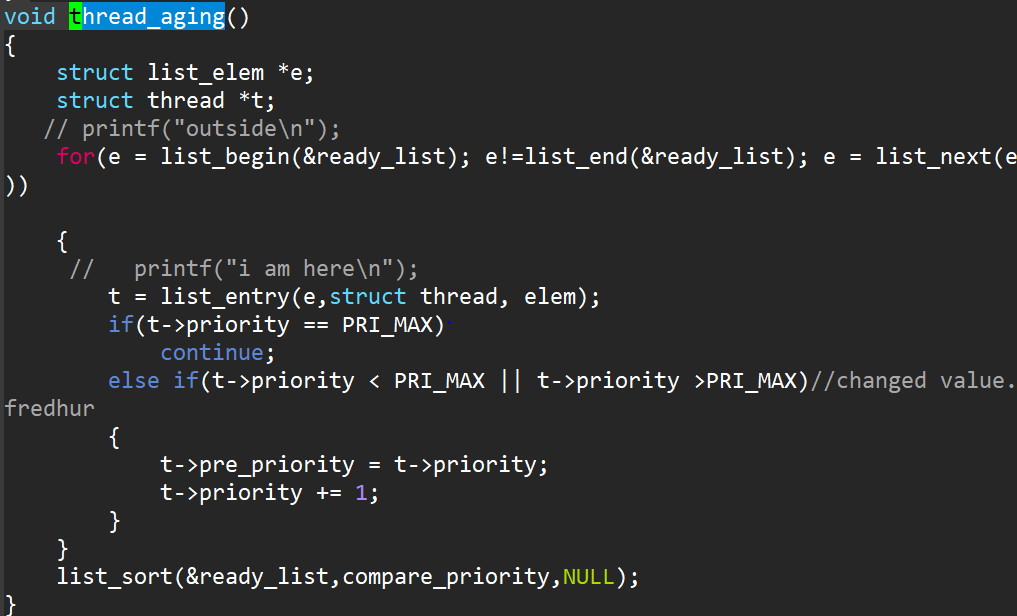
thread라면 idle thread인지 체크하고 아니라면 priority를 계산하여 설정해주는 함수이다.



Nice 변수를 계산하고 현재 쓰레드에 삽입해준다.

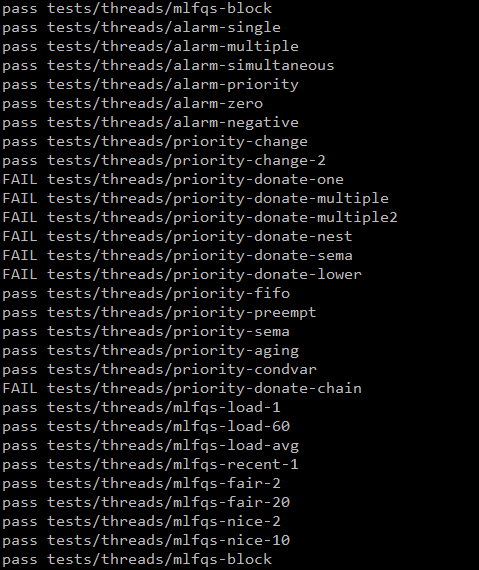
ASSERT는 if문과 비슷한 쓰임새로 block없어 사용할수 있다는 점에서

이번 프로젝트에서 유용하게 쓰였다.

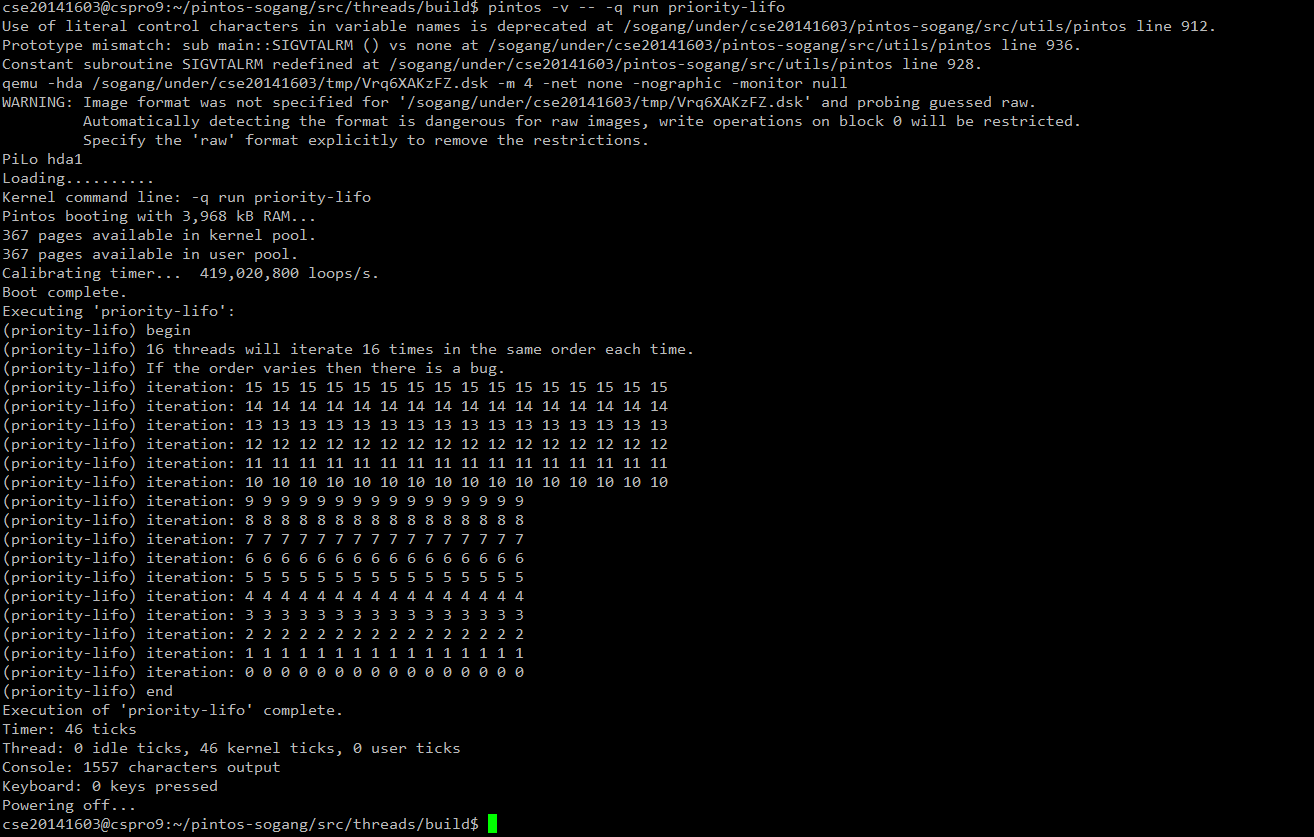


Thread\_aging 구현을 위한 함수 구성이다.

* 1. **시험 및 평가 내용**
* **ㅡmake check 부분**



**Priority-lifo 실행 부분.**



1. **기타**
   1. **연구 조원 기여도**

* **황순 ( 50 % ) , 허준형 ( 50 % )**
  1. **소감**
* **황순 – 이번 프로젝트는 수업을 나간 중간고사 부분이었다. 지금까지 했던 프로젝트들은 배우고 있는 과정이었거나 배울 과정이어서 직접 공부해야하는 부분이 많았지만, 이번 것은 배운 부분이어서 따로 공부를 할 것이 적지는 않았어도 지난번에 비해서는 적었다. 그렇지만 사소한 부분에서 오류가 계속 나서 아쉬웠고, 욕심을 내서 구현하지 않아도 될 부분을 구현하려다가 실패한 것이 아쉬웠다.**

**허준형 – 공부하였던 쓰레드 스케줄링 기법을 적용해 볼 수 있는 기회였다. 특히 에이징 기법으로 starvation 을 해결한 것과 mlfq 부분이 어려웠다. Mlfq 를 막상 구현해보니, 상당히 좋은 기법이라는 것을 알게 됐다. 또한 스케줄링의 방식을 위한 우선순위가 여러가지 있다는 것을 알 수 있는 좋은 공부였다. 특히 이번에는 makefile 을 수정하여 컴파일도 해봤는데, 거대한 코드를 어떤식으로 컴파일하여 적용하는지 알 수 있어서 좋았다.**