Nachos Term Project #2-1

❑ Project Overview

- **Project #2-1은 nachos/proj1 디렉토리에서 빌드 및 실행할 것.**
- **과제 제출 관련**
  - **제출 마감일: (※LMS에 등록할 것. 개인 email로는 받지 않음)**
    - **① 화요일 분반: 2022.5.7(화)**
    - **② 수요일 분반: 2022.5.8(수)**
    - **③ 목요일 분반: 2022.5.9(목)**
  - **제출물:**
    - **① Project Report**
    - **② 소스 코드 전체 (nachos 폴더 및 하위 폴더 전체 압축)**

❑ Task 1 (20 Points) – Thread 관련 루틴 분석

- **Proj1의 Task 1에서는 2개의 쓰레드가 교대로 실행되는 예제를 수행하였다. 해당 예제 실행과 관련하여 2개의 쓰레드가 생성, 실행, 및 소멸되는 과정을 Nachos 코드를 이용하여 상세히 분석하시오.**
  - 각각의 과정에서 호출되는 메소드 및 해당 메소드에서 일어나는 동작 등 쓰레드의 생성에서 소멸에 이르는 전 과정에 걸쳐 Nachos의 Class 및 Method Call Sequence와 동작 내용을 상세히 포함할 것
  - 예를 들어, 단순히 "현재 쓰레드의 실행을 멈추고 다음 쓰레드를 실행시킨다"로 작성하는 것이 아니라 "첫번째 쓰레드가 TCB.contextSwitch를 호출함. TCB.contextSwitch에서는 XXX 동작이 수행되며, 그 결과 다음에 실행될 쓰레드를 반환함"과 같이 **세부적으로 기술할수록 높은 점수**를 받을 수 있음
  - 전체 동작 과정을 이해하기 쉽도록 텍스트로만 작성하는 것보다는 시각적으로 표시할 것

❑ Task 2 (20 Points) – Alarm, Timer, Interrupt의 관계 및 동작 분석

- **Alarm, Timer, Interrupt 관련 소스 코드 내용을 확인하여 이들이 상호 연동 또는 동작하는 과정을 상세히 분석하시오**
  - Task 1과 같이 상세히 분석할수록 높은 점수를 받을 수 있음
  - 전체 동작 과정을 이해하기 쉽도록 텍스트로만 작성하는 것보다는 시각적으로 표시할 것

❑ Task 3 (50 Points) – Alarm::waitUntil(long x) 구현

- **Alarm::waitUntil() 메소드는 현재 시간에서 x time이 경과할 때까지 메소드를 호출한 쓰레드를 block시킴 (단, x는 Timer tick 단위로 표시)**
- **x Timer tick이상이 경과되면 해당 쓰레드는 ready 상태로 전이됨**
- **구현 가이드**
  - waitUntil() 메소드 및 timer 인터럽트 핸들러만 수정하면 됨
  - block된 쓰레드들의 리스트를 만들고 timer 인터럽트 발생할 때마다 timer tick이 x만큼

지났는지 검사 후 만약 지났을 경우 해당 쓰레드를 wake-up 시킴

- x Timer tick이 0 또는 음수인 경우 기다리지 않고 즉시 리턴함

❑ **Task 4 (10 Points) – 테스트 루틴을 Thread::selfTest()에 구현**

- **임의의 쓰레드를 생성 후 waitUnit() 호출 전 시간과 x timer tick이 지난 후 쓰레드가 wake-up된 후 실행될 때 시간을 출력하도록 코드 개발**
- **Hint: alarmTest.java**