

<Operating System>

## Assignment #3

“mytread”

소프트웨어학과

201321005

박주원

## 1. 구현한 것

과제를 진행하기 위해 User-level thread Library를 구현해야 했다. User-level threading은 커널에 의해 관리되는 kernel thread와 다르게 application 레벨에서 관리되는 thread library이다. 라이브러리 형태로 구현되므로 kernel 단에서는 user-level thread에 대해서 알 수 없다. User-level threading을 하는 이유는 Kernel thread context switching에 비해 비용이 저렴하고, Porting이 쉽다는 장점이 있기 때문이다. 단점으로는 I/O와 같은 kernel thread를 block 시키는 작업은 멈추게 된다. 최근 많은 언어들에서 light-weight user-level threading을 적극적으로 도입하고 있다.

1:n thread model의 user-level thread library를 작성했다. 해당 라이브러리를 가져다 쓰는 유저는 다음에 소개할 함수들을 활용해 threading을 사용할 수 있다. 두 가지 scheduling을 구현했는데, First-in First-out과 Round Robin 기법을 구현했다. 구현한 함수 중에 mythead 함수에서 mythead\_init();을 통해 mythead를 위한 모든 초기화를 하는 함수를 구현하고, scheduling 방식을 정해주면 해당 방식으로 scheduling이 된다. Static int nextJob()을 통해 정해진 scheduling 방식에 따라 다음 Job을 return 해 준다. Static void tick()은 scheduling 주기인 time quantum 마다 호출되는 함수이다.

## 2. 배운 것

과제를 진행하기 위해 p\_thread 라이브러리들의 기능에 대해 우선 공부が必要했다. 과제를 진행하면서 p\_thread 라이브러리들에 대해 확실히 알게 된 계기가 되었고, FIFO와 ROUND ROBIN scheduling을 직접 구현하면서 그저 보고 이해만 했던 스케줄링 기법들에 대해 자세히 알 수 있게 되었다. Thread를 생성하고 스케줄링하는 과정을 통해 전체적인 구조에 대해 다시금 생각해보게 되는 계기가 되어 많은 것을 배우게 된 것 같다. 과제를 진행하기 위해 문제를 이해하고 소스코드 구조를 이해하는 것이 중요하다는 것을 깨닫게 해주었고, ucontext에 관련된 system call들에 대해 알게 되었다.

### 3. 피드백

이번 과제는 스켈레톤 코드에 어느 정도 코드에 대한 주석이 달려있어서 과제를 진행하는데 많은 도움이 되었다. Mysh-0과 mysh-1 과제보다 훨씬 더 구현하기 어려웠지만 문제를 이해하고 소스코드 구조를 이해하면 비교적 구현이 간단하다는 조교님의 말씀대로 문제를 먼저 이해하는 것이 가장 중요했던 것 같다. 이번 mythread 과제는 운영체제라는 과목에서 가장 필요하고 중요했던 과제였던 것 같다. 교수님이 수업시간에 가르치신 내용들에 대해 이렇게 직접 구현해보니 이해가 완전히 되는 그런 느낌이었다.

### 4. 튜토리얼

- 1) Main.c 에서 user-level thread가 작업할 void f()와 void g() 함수 작성  
Ex) f : A ~ J 까지 1초 간격으로 출력  
g : 1부터 입력 받은 n까지 1초 간격으로 출력
- 2) 실행 후 Scheduling 기법을 정한다. (1을 입력하면 FIFO, 2를 입력하면 RR)  
Ex) FIFO를 선택한 경우  
`Select Scheduling Type(1.FIFO, 2.RR) : 1`
- 3) n을 사용자로부터 입력받는다.  
Ex) `Enter the number of n (ex: 8): 8`
- 4) Mythread\_create 함수를 통해 스레드를 생성한다.
- 5) Mythread\_join 함수를 통해 스레드를 기다린다.

Example

n = 8, Round Robin 선택했을 시

```
Select Scheduling Type(1.FIFO, 2.RR) *Enter 1 Or 2 : 2
Enter the number of n (ex: 8): 8
Scheduling : Round Robin
f : A
f : B
g : 1
g : 2
f : A
f : B
g : 1
g : 2
Waiting thread 1
f : C
f : D
g : 3
g : 4
f : C
f : D
g : 3
g : 4
f : E
f : F
g : 5
g : 6
f : E
f : F
g : 5
g : 6
f : G
f : H
g : 7
g : 8
f : G
f : H
g : 7
g : 8
f : I
f : J
Thread 1 Finished
Waiting thread 2
Thread 2 Finished
Waiting thread 3
f : I
f : J
Thread 3 Finished
Waiting thread 4
Thread 4 Finished
```