**OS Project Report 2: Priority Scheduling**

20205031 김소정, 20205135 이수린, Team F(team 5)

**Problem1-1: Implementing Priority Scheduling**

1. **Problem definition**

현재 pintos 운영체제에서는 스레드 스케줄링을 라운드 로빈 방식으로 수행한다. 이 방식은 스레드를 ready\_list의 뒤에 추가하고, 맨 앞에서 제거한다. 이러한 단순한 방식은 스레드의 우선순위를 고려하지 않으며, 결과적으로 자원 활용의 비효율성과 작업 실행의 불공평성을 초래할 수 있다.

1. **Policy and algorithm design**

* Policy
  + Priority scheduling: 우선순위가 높은 스레드에게 CPU를 할당하여 시스템의 응답성을 향상시키고, 우선순위가 높은 작업을 빠르게 처리하는 것을 목표
* Algorithm design
  + thread\_create(): 새로 생성된 스레드가 현재 실행중인 스레드가 현재 실행 중인 스레드보다 우선순위가 높을 경우 CPU를 선점하도록 함. 이를 위해 새로 생성된 스레드의 우선순위와 현재 스레드의 우선순위를 비교하여 스케줄링 수행
  + list\_insert\_orderd() 시용: 리스트에 새로운 요소를 우선순위에 따라 정렬하여 삽입하는 함수. Ready\_list에 새로운 스레드가 삽입될 때 우선순위에 따라 정렬된 위치에 삽입
  + 스레드 생성 및 준비 상태로 전환: 새로운 스레드가 생성되거나 블록 상태에서 준비 상태로 전환될 떄, ready\_List에 삽입될 때 우선순위에 따라 정렬된 위치에 삽입
  + 스레드 양보 및 스케줄링: 현재 실행 중인 스레드가 실행을 양보할 때, ready\_list에 새로운 스레드를 추가할 떄, ‘list\_insert\_ordered()’ 함수를 사용하여 우선순위에 따라 정렬된 위치에 삽입되도록 하고, ‘thread\_yiedl()’ 함수를 호출하여 CPU를 양보하여 스케줄링이 수행

1. **Mechanism**

* ADD for Priority Scheduling: compare\_thread\_priority 함수, change\_thread\_priority 함수

Modified for Priority Scheduling: thread\_unblock(), thread\_yield(), thread\_create(), thread\_set\_priority()

* Modified file names
  + /home/name/pintos/src/threads/thread.c
  + /home/name/pintos/src/threads/thread.h
* CODES
  + Compare\_thread\_priority 함수 구현

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명두 스레드의 우선순위를 비교하여 높은 우선순위를 가진 스레드가 먼저 실행되도록 함. 리스트에서 스레드를 우선순위에 따라 정렬될 때 사용.

* + 텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명Thread\_unblock(), thread\_yield() 함수 수정

새로운 스레드가 ready\_list에 삽입될 때 ‘list\_insert\_ordered’ 함수를 사용하여 우선순위에 따라 정렬된 위치에 삽입

* + Change\_thread\_priority() 함수 구현

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

현재 실행 중인 스레드의 우선순위가 변경된 경우를 감지하고, 필요한 경우에 스레드의 우선순위를 변경함. 이 함수는 ready\_list의 가장 높은 우선순위를 가진 스레드와 현재 스레드의 우선순위를 비교하여, 현재 스레드의 우선순위가 더 낮으면 CPU를 양보하여 스케줄링이 수행

* + Thread\_create(), thread\_set\_priority() 함수 수정

현재텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명 실행중인 running 스레드의 priority가 바뀌는 경우를 고려

Thread\_create(): 새로운 스레드가 생성되었을 때, change\_thread\_priority 함수를 호출하여 필요한 경우에 스레드의 우선순위를 변경

Thread\_set\_priority(): 현재 스레드의 우선순위를 새로운 우선순위로 변경하고, 변경된 우선순위가 이전 우선순위와 다르거나 새로운 우선 순위가 이전 우선순위보다 높은 경우에는 스레드의 우선순위를 업데이트하고 스레드의 스케줄링 조정

**Problem1-2: Implementing Donation**

1. **Problem definition**

Priority inversion 문제는 priority가 높은 스레드가 priority가 낮은 스레드를 기다리는 상황이 발생하면서 일어난다. 이로 인해 시스템의 성능 저하 및 우선순위 역전 현상이 발생할 수 있다.

1. **Policy and algorithm design**

* Policy
  + Priority donation: priority 가 높은 스레드가 자신보다 우선순위가 낮은 스레드를 기다리는 경우, 자신의 우선순위를 기다리는 스레드에게 기부
  + Multiple donation: 스레드가 여러 개의 락을 보유하고 있는 경우, 각 락을 획득하고자 하는 스레들에게 기부가 발생할 수 있음. 이를 위해 스레드는 이전에 기부한 우선순위를 기억하고 락을 획득한 후에는 해당 락을 기다리는 스레드들에게 우선순위를 재할당
  + Nested donation: 높은 우선순위를 가진 스레드가 우선순위가 낮은 스레드들에게 기부되어야 함. 다른 스레드들이 여러 락을 보유하고 있을 때, 우선순위가 높은 스레드가 순차적으로 락을 해제하여 가장 높은 우선순위를 가진 스레드가 정상적으로 락을 획득할 수 있도록 함
* Algorithm design
  + 우선순위 기부를 위한 구조체 추가: 추가적인 정보를 스레드 별로 저장해야 함. 스레드의 원래 우순위, 본인이 기부를 한 lock, 본인이 들고 있는 lock 들의 정보, thread 구조체에 추가적으로 선언하고 lock로 list에 들어가야하므로, lock 구조체에도 이를 위한 list\_elem 선언
  + Donate\_priority() 구현: 현재 스레드가 대기 중인 락에 대해 우선순위를 기부함. 대기 중인 락에 대한 우선순위를 기부하고, 해당 락의 소유자 스레드의 우선순위도 재귀적으로 확인하여 필요한 경우 우선순위를 업데이트
  + Update\_priority() 구현: 기부된 우선순위를 복구하고, 기부한 스레드의 리스트에서 해당 스레드를 제거. 락을 해제할 때 이전 우선순위를 확인하여 기부한 우선순위가 더 이상 필요하지 않은 경우 본래의 우선순위로 복구
  + Lock\_acquire, lock\_release 함수 수정: 락을 획득 및 해제할 때마다 우선순위 관리 및 기부 과정을 수행하여 priroiryt donation 정책을 구현

1. **Mechanism**

* ADD for Priority Donation: donate\_priority(), update\_priority(), sort\_ready\_list()

Modified for Priority Donation: lock\_acquire(), lock\_release(), thread\_set\_priroity(), sema\_up()

* Modified file names
  + /home/name/pintos/src/threads/thread.c
  + /home/name/pintos/src/threads/thread.h
  + /home/name/pintos/src/threads/synch.c
  + /home/name/pintos/src/threads/synch.h
* CODES
  + 스레드 구조체에 추가 변수 선언 및 초기화 함수 수정 & lock 구조체에도 스레드의 리스트에 넣을 수 있도록 ‘list\_elem’ 추가
  + Lock\_acquire 수정  
    현재텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명 스레드가 해당 락을 이미 소유하고 있는지 확인, lock의 소유자를 확인하고, 만약 소유자가 이미 있다면, 해당 락을 기다리는 스레드의 우선순위를 기부(donate).   
    sema\_down 함수를 사용해 락을 획득하고, 락을 획득한 후에는 현재 스레드를 락의 소유자로 설정하고, 현재 스레드의 락 리스트에 해당 락을 추가
  + Sort\_ready\_List 구현

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명준비 리스트를 스레드의 우선순위에 따라 정렬, 가장 높은 우선순위를 가진 스레드가 맨 앞에 위치

* + Donate\_priority 구현

현재텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 스레드가 특정 락을 기다리는 동안 다른 스레드에게 우선순위를 기부:  
이 과정에서 thread\_ready 상태에 있는 스레드의 우선순위가 변경되면, is\_ready 플래그를 설정하여 후속 작업에서 준비리스트를 업데이트 해야함을 나타냄. 마지막으로 sort\_ready\_list 함수를 호출하여 준비 리스트를 업데이트

* + 텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

    자동 생성된 설명Sema\_up 수정

세마포어의 waiters에서 기다리고 있는 스레드들의 우선순위 또한 변할 수 있음. 따라서 함수 내에서 스레드를 뽑기 전에 정렬. 스레드의 우선순위가 변경되었고, 우선순위가 변경된 스레드가 있을 경우 thread\_yield 호출

* + Lock\_release 수정  
    현재텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명 스레드가 해당 락을 소유하고 있는지 확인 후 락의 소유자를 NULL로 설정하여 해당 락을 해제, 현재 스레드의 락 리스트에서 해당 락을 제거. 락의 세마포어를 sema\_up 함수를 사용해 증가시켜 다른 스레드가 이 락을 얻을 수 있도록 함. 락 해제 이후에는 update\_priority 함수를 호출하여 우선순위를 업데이트
  + Update\_priority 구현

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

이 함수는 현재 스레드의 우선순위를 업데이트하는 데 사용

함수는 현재 스레드가 보유한 모든 락의 대기열을 검사하여 가장 높은 우선순위를 결정, 락의 대기열이 비어있지 않은 경우, 대기열에서 가장 높은 우선순위를 가진 스레드의 우선순위를 확인. 현재 스레드의 우선순위를 해당 락의 대기열에서 가장 높은 우선순위로 업데이트

* + Thread\_set\_priority 수정   
    텍스트, 스크린샷, 폰트이(가) 표시된 사진

    자동 생성된 설명

현재 스레드의 우선순위를 설정하고 이 함수는 새로운 우선순위가 현재 우선순위보다 높은 경우에만, 우선순위를 업데이트. Original\_priority를 업데이트하고, 우선순위를 변경한 경우 change\_thread\_priority 함수를 호출하여 스레드의 우선순위를 변경