

COM2005 Operating Systems - Term Project

Shortest Remaining Time First (SRTF) 스케줄러 구현하기

과제 개요

- CPU 스케줄러는 프로세스의 동작을 제어하며 멀티태스킹(multitasking)이 가능케 만듭니다. CPU 스케줄러는 프로세스가 언제, 어떻게 CPU를 사용할지 결정합니다. 스케줄러를 통해, 시스템에서 동작하는 여러 프로세스가 CPU를 공평하게 사용할 수 있으며, 시간 제약이 있는 프로세스가 적절한 타이밍에 CPU를 사용할 수 있습니다.
- 이 과제의 목표는 가상 컴퓨터 환경에서 동작하는 스케줄러를 만드는 것입니다. 가상 컴퓨터 환경을 구축하기 위한 코드는 과제와 함께 제공됩니다. 리눅스 커널이 스케줄링에 사용하는 API를 가상 컴퓨터 환경에서도 동일하게 사용할 수 있습니다. 커널에서 프로그래밍할 때와 다른 점은 표준 C함수를 사용할 수 있다는 점입니다. 스케줄링 구현과 관련된 구조체와 매크로를 모두 포함하고 있기 때문에, 제공된 코드만을 충분히 구현할 수 있을 것입니다.
- 실제 리눅스 스케줄러는 매우 복잡하기 때문에 가상 컴퓨터를 제작하게 되었습니다. 리눅스 커널의 스케줄러는 프로세스의 다양한 동작을 모두 처리할 수 있어야 하고, 멀티프로세서(multiprocessor)를 지원해야 하며, 이에 따른 메모리 관리도 해주어야 합니다. 이런 복잡한 기능보다는 스케줄러의 동작에 집중할 수 있도록 단일 프로세스 환경으로 가상 컴퓨터 환경을 제작하였습니다. 리눅스 커널의 원본 코드를 바탕으로 제작되었기 때문에 동작은 거의 동일합니다.

과제 수행 시 주의 사항

- 제공된 파일 중 `schedule.c`와 `schedule.h`를 작성해야 합니다. 각 함수마다 스케줄러에서 무슨 역할을 하는지 주석이 달려있습니다. `schedule.h` 파일에는 스케줄러를 구현하기 위해 필요한 함수의 목록이 작성되어 있습니다. 가상 컴퓨터는 이 함수들을 바탕으로 작업을 수행합니다. (가급적 리눅스 커널의 `kernel/sched.c`, `kernel/sched.h` 파일을 열어보고 어떻게 동작하는지 참조하길 권해드립니다. 코드의 양이 많기는 하지만, 각 함수에서 해야 할 일에 대한 아이디어를 얻을 수 있을 것입니다.)
- 가상 컴퓨터에서 제공되는 시스템콜에 어떤 것들이 있는지 이해해야 합니다. 시스템콜은 스케줄러를 작성할 때 유용하게 사용할 수 있습니다. 시스템콜은 `schedule.h`에 작성되어 있습니다.
- `schedule.h` 파일에 있는 데이터 구조체를 눈여겨보시기 바랍니다. 이 구조체는 리눅스

커널에 존재하는 구조체를 단순화한 것입니다. 단순화하기는 했지만 동작은 동일합니다.

- 원하는 순서대로 함수를 구현해도 상관없지만, 일부 함수는 초반에 구현하는 것이 좋습니다. *initschedule*과 *killschedule*은 가상 컴퓨터에서 스케줄러 초기화와 종료기능을 담당하며, 가상 컴퓨터가 정상적으로 동작하기 위해 반드시 필요합니다. 그 다음 중요한 함수는 *activate_task*와 *deactivate_task*입니다. *schedule*은 스케줄러의 핵심 기능을 담당하므로 초기에 구현하는 것이 좋습니다.
- 어느 정도 코딩이 완료되었다면 가상 컴퓨터와 함께 컴파일할 수 있을 것입니다. 완성된 코드는 커맨드 라인을 통해 실행할 수 있습니다. 가상 컴퓨터는 프로세스의 프로필에 따라 언제 생성 시기와 생성 방법을 결정합니다. 올바른 테스트를 위해 샘플 프로필을 함께 제공하니, 이를 이용해 테스트해보기 바랍니다. 물론 본인이 직접 프로필을 작성해서 사용해도 됩니다.

제출 방법

- 제출할 것: *schedule.c*, *schedule.h*, *Makefile*, 추가로 생성하거나 수정한 파일
- 제출 방법: 교과목 블랙보드를 통해 제출
- 제출 기한: 2019년 6월 23일(일)