Operating Systems

(Scheduler & Thread)

Spring, 2020



- Understanding
 - 1. Thread programming
 - 2. Thread synchronization
 - 3. Scheduler simulation



프로그래밍 환경

- pthread library 를 사용해야 하니 리눅스 환경에 구축되어야 합 니다.
 - 아래 교재 사이트에 virtual machine 설치가 자세히 나왔습니다. 참조하 시면 됩니다.
 - http://cs.westminstercollege.edu/~greg/osc10e/vm/index.html
 - 상기 환경에서 교재의 예제 프로그램도 확인할 수 있습니다.



가상 시스템 가져오기 가져올 가상 시스템 Please choose the source to import appliance from, This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from, Source: Local File System Please choose a file to import the virtual appliance from, VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below, 파일(F): C:₩Users₩msson₩Documents₩course₩OS_2020_강좌₩OSC10e,ova

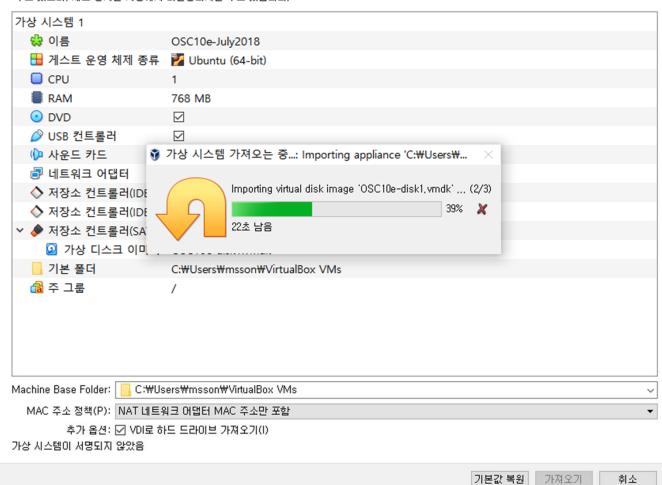
전문가 모드(E) 다음(N)

취소

← 가상 시스템 가져오기

가상 시스템 설정

아래 목록은 가상 시스템 설명 파일에 나와 있는 가상 머신이며, 이를 VirtualBox로 가져왔을 때의 형태입니다. 보여져 있는 속성을 두 번 누르면 변경할 수도 있으며, 체크 상자를 사용해서 비활성화시킬 수도 있습니다.



```
7 OSC10e-July2018 [실행 중] - Oracle VM VirtualBox
                                                                                                                                                                                                                                                                   파일 머신 보기 입력 장치 도움말
Jbuntu 16.04.5 LTS ubuntu tty1
ubuntu login: osc
Password:
 Last login: Sat Aug 11 13:47:56 MDT 2018 from 10.0.2.2 on pts/0
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-87-generic x86_64)
   * Documentation: https://help.ubuntu.com
   * Management:
                                                     https://landscape.canonical.com
                                                    https://ubuntu.com/advantage
   * Support:
182 packages can be updated.
173 updates are security updates.
>> Welcome to the Linux VM for Operating System Concepts!
osc@ubuntu:~$ Is
   inal-src-osc10e final-src-osc10e.zip README
osc@ubuntu:~$ cd final-src-osc10e/
osc@ubuntu:~/final-src-osc10e$ ls
osc@ubuntu:~/final-src-osc10e$ cd ch3
osc@ubuntu:~/final-src-osc10e/ch3$ ls
DateClient.java fig3-32.c multi-fork
                                                                                                                                pid.c
                                                                                                                                                                                               unix_pipe.c
DateServer.java fig3-33.c multi-fork.c
                                                                                                                                shm-posix-consumer.c win32-pipe-child.c
                                               fig3-34.c newproc-posix.c shm-posix-producer.c win32-pipe-parent.c
 fig3-30.c
                                               fig3-35.c newproc-win32.c simple-shell.c
fig3-31.c
osc@ubuntu:~/final-src-osc10e/ch3$

    O III I P O III I
```

🛂 OSC10e-July2018 [실행 중] - Oracle VM VirtualBox X 파일 머신 보기 입력 장치 도움말 osc@ubuntu:~/final-src-osc10e/ch4\$ ls Driver.java fig4-23.c openmp.c SumTask.java thrd-posix.c thrd-win32.c ThreadPoolExample.java osc@ubuntu:~/final-src-osc10e/ch4\$ gcc -pthread thrd-posix.c -o thrd_test osc@ubuntu:~/final-src-osc10e/ch4\$ ls Driver.java openmp.c thrd-posix.c thrd-win32.c SumTask.java thrd_test fig4-23.c ThreadPoolExample.java osc@ubuntu:~/final-src-osc10e/ch4\$./thrd_test 3 sum = 6osc@ubuntu:~/final-src-osc10e/ch4\$_

Thread programming

• 아래 예제 프로그램을 확인하기 바랍니다.(4장 강의 참조)

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
int sum; /* this data is shared by the thread(s) */
void *runner(void *param); /* the thread */
int main(int argc, char *argv[])
  pthread_t tid; /* the thread identifier */
  pthread_attr_t attr; /* set of attributes for the thread */
  /* get the default attributes */
  pthread_attr_init(&attr);
  /* create the thread */
  pthread_create(&tid,&attr,runner,argv[1]);
  /* now wait for the thread to exit */
  pthread_join(tid, NULL);
  printf("sum = %dWn",sum);
```



Thread synchronization

■ POSIX semaphore 활용하기 바랍니다. (6, 7장 강의 참조)

```
#include <semaphore.h>
sem_t sem;
sem_init, sem_wait, sem_post 함수
```



Scheduler

- 5장에서 다양한 스케줄러에 대해서 학습하였으니, 참조하기 바랍니다.
 - SJF
 - Round-Robin
 - Priority
 - Completely Fair Sharing (CFS)
 - Multi-level queue

What to do?

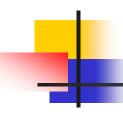
- 1. 본인만의 스케줄러를 설계하고 시뮬레이션하시오.
 - 자유 과제임
 - (자유과제기 때문에 다른 학생 숙제 copy 불가함 → 유사한 경우 별도 정밀 평가 가 있을 예정이니, 절대 보여주거나 copy 하지 말기 바람)
 - 난이도에 따라서 점수가 달라짐
 - 본인의 실력에 맞추어서 설계를 잘 하기 바람
 - 보고서를 잘 작성하기 바람
 - 보고서에 대한 면밀한 평가가 있을 예정임 (기승전결 잘 기술하기 바람)
 - 보고서 기술이 잘 안될 경우 구현하더라도 점수가 미미함
 - 본인의 스케줄러를 강조하면서 설계하기 바람
 - 어떻게 동작하는지, 각각의 큐에서 어떻게 스케줄링이 되고, 큐 간의 스케줄링이 무엇 인지 명시바람
 - 자유 과제로 어떻게 결과가 나오는지를 명확하게 기술하기 바람
 - 쓰레드 동기화에 대해서 어떻게 사용했는지 명확하게 기술하기 바람



- 2. 스케줄링은 별도의 쓰레드에서 수행하기 바람
 - 단순히 1개의 스케줄링 (난이도 하) 을 구현하더라도 별도의 쓰레드에서 구현하기 바람
 - multi-level queue 기법 사용 시 (난이도 중)에는 반드시 별도의 쓰레드 를 생성하고, 각 queue 안의 스케줄링은 별도의 쓰레드에서 처리하기 바람
 - multi-level queue 에서 큐간의 스케줄링은 semaphore 를 활용해서 동기화하기 바람
 - 큐 간의 스케줄링을 담당하는 쓰레드를 두기 바랍니다.
 - Multi-level queue 의 개수는 5개로 제한을 둡니다.
 - 추가적인 본인의 스케줄링 또는 복잡한 스케줄링 (난이도 상) 이 있을 경우에는 해당 내용을 보고서에서 강조하기 바람

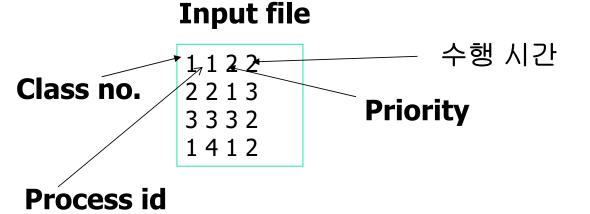


- 3. Input file 제공 바람
- 4. 어떻게 동작하였는지를 기술하기 바람
- 5. 제출 내용
 - 보고서, input file, 소스 코드 등
 - 컴파일 (gcc) 사용



어떻게 스케줄링 워크로드를 생성하는가?

4가지 input (Class no. process id priority 수행 시간)



- 1. Process id 는 모두 달라야 함
- 2. Priority 스케줄링이 안쓰일 때는 priority 는 무시됨 (0으로 기입)
- 3. 기타 스케줄링 방법이 있을때는 define 을 보고서에 잘하기 바람 (예, RR 에서 quantum, CFS 에서 priority 1 은 비율 weight 얼마) 🕫

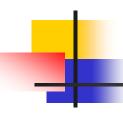


- 간단한 예제
 - 1> 총 3개의 큐
 - 2> 큐 안의 스케줄링
 - 1번째 큐 : 우선순위
 - 2번째 큐 : 라운드로빈
 - 3번째 큐 : SJF
 - 3> 큐 간의 스케줄링
 - Priority 스케줄링 (1번째 큐 > 2번째 큐 > 3번째 큐)

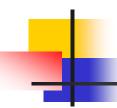
결과 파일 (서로 다른 id 는 다른 줄에 기술)

1122 2213 3 3 3 2 1412





- 주의 사항
 - Input file 에서 모든 태스크는 0 에 동일하게 도착한다고 가정함
 - 샘플 input file 과 output file 제시하기 반드시 바람
 - 본인의 아이디어를 잘 드러낼 수 있는 파일이어야 함



No submission after the deadline

Deadline: May 29th 23:59

Second-chance deadline: May 30th 23:59

• 원래 데드라인은 5월 29일이나, 서버 이상으로 인한 미제출을 방지하기 위해서 5월 30일까지 기회는 주나, 5월 30일 23:59 분 이후는 어떤 경우에도 절대 제출 불가이며, 어떤 경우도 받지 않음