

# Operating Systems

## 1. 강의 정보

- 운영체제 (Operating Systems) - 2150516201, 2150516202(전필, 소중필, 설계과목)
- 홍지만 (정보관 525호, jiman@ssu.ac.kr, 02-828-7168) 월/수 14:00~16:00 - 이메일 또는 전화로 상담 신청
- 수업을 위한 기본 준비
  - 강의 게시판 : classroom.google.com/
    - ✓ 대학에서 기능이 훌륭한 강의 게시판 또는 유료 구글클래스룸 구입 시 변경 가능
    - ✓ 기존 사용 구글 이메일 아이디(계정) 또는 새로운 구글 이메일 아이디 생성 (**새로운 이메일 계정을 만드는 것을 권장**, 게시판 게시 내용 변경 사항, 새로운 강의노트, 질문에 대한 응답 등이 해당 이메일로 보내질 것이니 **새로운 이메일 아이디를 사용할 경우 이메일을 자주 확인해야 함**)
    - ✓ 스마트캠퍼스(LMS)를 사용하지 않고 구글클래스룸을 사용하는 이유는 질문에 대한 빠른 피드백 가능(과제 및 강의 관련 질문 시 바로 이메일로 확인이 가능하고 이메일로 답변을 할 수 있음)하고 기타 장점이 더 많기 때문임. 스마트캠퍼스와 구글클래스룸을 모두 사용해야 하기 때문에 다소 불편할 수 있으나, 본 수업 수업을 위해서는 불편함을 감수해야 함.
    - ✓ 수업 시작 후 몇 주간은 공지 사항과 추가 강의노트 등을 스마트 캠퍼스에도 게시할 예정임.
    - ✓ 구글클래스룸에서 사용 가능한 아이디는 "**분반.이름.학번(두자리 수)\_학번뒤4자리**" (예. 가\_홍길동\_14\_0001)
    - ✓ 구글클래스룸에서 사용 가능한 아이디는 수업참여도(질문) 등을 확인과 성적 처리를 위해 반드시 아이디 명명 규칙에 따라 만들어함
      - 구글클래스룸에서 사용할 아이디는 반드시 설정에서 변경
      - 구글클래스룸 아이디는 구글 이메일 아이디 설정에서 변경을 하면 자동으로 변경되기 때문에 아이디 변경 주의
      - 다른 사람에게 이메일을 보낼 때 설정된 아이디로 이메일이 갈 수 있음을 주의. 이메일로 등록된 아이디가 갑자기 변경된 경우 학생의 정보를 알 수 없기 때문에 과제 채점이 불가능할 수 있음. 지속적으로 아이디를 변경하여 과제 채점이 어려울 경우 F 학점 부여
      - 구글클래스룸 등록된 이메일은 자주 확인해야 함도 주지하기 바람.
    - ✓ classroom.google.com -> 오른쪽 상단 (+) 클릭 -> 수업 코드 q4btj6f -> 참여하기
      - <https://classroom.google.com/u/0/c/NjE2OTMyMDM3NjU1>
      - 과제 제출 : (상단) 수업 탭 -> 과제 번호 선택
      - 수업 관련 질문 : (상단) 스트림 탭 -> 질문 (소식이나 자료 등을 공유해 보세요. 클릭)
      - 과제 관련 질문 : (상단) 수업 탭 -> 과제 선택 -> 과제 보기 메뉴 -> 수업 댓글 추가
- 수업시간 및 강의실
  - (가)반 : 월 10:30~11:45 / 수 12:00~13:15, (나)반 : 월 12:00~13:15 / 수 10:30~11:45
  - ✓ 신청한 강의의 수업 시간만 수강할 수 있으며 다른 반 수강은 출석 인정하지 않음
  - ✓ 한글날, 출장, 추석연휴전날, 추석연휴다음날 등 사전녹화강의는 자동 출결 점검을 위하여 스마트 캠퍼스를 이용. 출석
  - ✓ 출석기한은 다음 수업일 전날 밤 12시까지.
- 교재
  - Operating Systems: Three Easy Pieces, Remzi, 2018,
    - ✓ [https://www.academia.edu/41994710/OPERATING\\_SYSTEMS\\_THREE\\_EASY\\_PIECES](https://www.academia.edu/41994710/OPERATING_SYSTEMS_THREE_EASY_PIECES)
  - Operating Systems, Internals and Design Principles, 8th ed, by William Stallings
  - Modern Operating Systems, 3<sup>rd</sup> ed, by A. Tanenbaum
  - Operating Systems Concepts, 10<sup>th</sup> ed, by A. Silberschatz, P. Galvin and G. Gagne
  - 기타 강의 노트 (필요시 강의 전 구글클래스룸을 통해 배포)

## 2. 교과목 개요

- 본 교과목에서는 범용 운영체제의 개념, 기능, 기본 동작 원리를 이해하고 실제 운영체제 시스템을 설계하고 구현할 수 있는 능력을 키움

3 Easy Piece (UW)	Abstracted by struct(class)	HW
Virtualization	Process Structure (task_struct)	CPU

	Memory Management (mm_struct ->vm_area struct ->)	Memory
Concurrency	Process (semaphore, thread...)	CPU
Persistent	File System (files_struct -> file struct-> inode struct)	DISK

### 3. 교과목 교육 목표

- 본 교과목을 이수함으로써, 다양한 운영체제 상에서 시스템 소프트웨어를 구현하는 능력과 임베디드 운영체제를 이해하고 구현하는 기본 능력을 배양할 수 있다.
  - 1. 운영체제 설계 및 구현 시 필요한 이론과 자료 구조를 이해하는 능력을 배양한다.
  - 2. 운영체제와 그와 관련된 시스템 소프트웨어를 설계하고 구현할 수 있는 능력을 배양한다.
  - 3. 실무에서 운영체제의 특성에 의해 발생하는 문제를 인식하고 이를 해결할 수 있는 능력을 배양한다.

### 4. 강의 방법 및 과제

- 원서와 자체제작 교재(ppt)를 활용하여 이론 강의
  - 학생들은 반드시 수업시간에 교재를 반드시 지참해야 함
    - ✓ PART1 소개
    - ✓ PART2 Virtualization (CPU 가상화, 스케줄링, 메모리 가상화)
    - ✓ PART3 Concurrency (세마포어, 스레드, 데드락)
    - ✓ PART4 Persistence (File System, I/O)
- 설계과제
  - 오픈소스 기반 운영체제(xv6) 의 설계 및 구현
    - ✓ 총 4~5개로 구성, 강의 일정표 참고. 과제 내용은 구글클래스룸 게시판을 통해 확인

### 5. 평가 기준 및 설계 과제


- 중간시험(30%), 기말시험(30%), 설계과제(35%), 출석(3%), 수업참가도(2%)
  - 중간시험 및 기말시험 : 각 100점 만점을 기준으로 각각 30%로 환산하여 총점에 반영 (중간시험 및 기말시험은 각 100점 만점 기준 총점에 각 30점씩 반영). 단, 최종 등급별 비율(가/나/반) 합반 처리 예정임. 등급별 비율은 추후 공지.
- 설계과제 제출 방법
  - 설계과제는 “#설계과제번호\_학번\_분반\_버전.zip”(예. #1\_20160000\_나\_V1.zip)형태로 (zip프로그램으로 압축하여) 파일 이름을 명명하고, 구글클래스룸에 제출해야 함.
    - ✓ 제출한 압축 파일 내 설계보고서 파일(#1\_20160000\_가\_V1.hwp 또는 #1\_20160000\_가\_V1.doc 형태)가 있어야 하며 **소스코드는 별도로 “소스코드” 디렉토리** 내에 있어야 함
    - ✓ 버전 이름은 대문자 V와 함께 integer를 1부터 incremental 증가시키면서 부여하면 됨. (예. V1, V2, V3..), 단 처음 제출 시는 버전 번호를 붙이지 않아도 되며 두 번째부터 V1를 붙여 제출하면 됨.
    - ✓ 설계과제 보고서 파일은 표지를 만들지 말고, 바로 첫 장, 첫 줄에 학과, 학번, 이름 정보를 오른쪽 정렬로 기입하고, 바로 밑에 1. **개요**(설계과제 설명서의 개요를 좀 더 상세하게 설명해야 함), 2. **상세설계**(구현한 함수 **프로토타입**, 구현한 내용과 기 구현된 함수 간의 **호출 그래프** 등 등 xv6에 기구현된 내용과 수정한 내용 등을 상세하게 설명하는 내용 등), 3. **결과**(다양한 입력에 따른 실행 결과 등을 캡처 및 해석 등 포함), 4. **주석 달린 소스코드** (수정한 소스코드와 makefile, obj, \*.c, \*.h 등 컴파일하고 실행하기 위한 모든 파일을 포함시켜야 함)를 반드시 포함시켜야 함. 글자 크기는 10pt, 왼쪽 여백 15pt, 오른쪽 여백 15pt, 줄 간격은 “글자에 따라 150%”로 할 것.
    - 글자 모양 등은 임의대로 정하면 됨. MS 워드를 사용할 경우에도 비슷하게 크기로 만들면 됨.
    - 보고서 양식은 위에서 지정한 크기와 비슷하게 만들면 됨. 정확하게 지키지 않아도 됨. 단, 표지를 포함시킨 경우, 정말 큰/작은 폰트 사용한 경우, 좌우/아래위 여백이 정말 큰/작은 경우 등 눈으로 봤을 때 지정한 양식보다 크기가 너무 차이 나는 경우 해당 과제 성적에서 10점 감점.
    - ✓ 설계과제 제출 방법을 따르지 않은 경우 해당 과제 성적에서 10점 감점.

- 설계과제는 마감 시간 이후 24시간까지 지연 제출 가능. 그 이후 제출은 0점 처리. 설계과제 마감 시간 이후 지연 제출은 30% 감점.

#### ○ 설계과제 배점 기준

- 기한 내에 여러 버전의 보고서를 제출 할 수 있으나, 채점은 최종 버전만을 대상으로 함.
- 과제 채점 시 첨부 파일이 없을 경우 과제 미제출로 처리 (예외 사항 없음)
- 건물 정전, 제출 서버시스템의 결함 등으로 인해 시스템 접속이 불가능할 경우 반드시 마감 기간 내 이메일 (2023.oslab@gmail.com)로 제출해야 함. 단 이메일 제목은 "P설계과제번호\_학번" 이어야 하고, 내용 없이 #P설계과제번호\_학번.zip만 첨부하면 됨.
- 설계과제는 과제당 기본점수 100점 만점 \* 총 4~5개 설계과제 = 400~500점 만점을 기준으로 35% 환산 성적 부여 (설계과제 총점은 100점 만점에 35점)
- 각 설계과제는 구현 완성도에 따라 (해당 과제 100점 만점 기준) 가산점 부여.
  - ✓ 단, 가산점 부여 대상이 되는 과제는 (1) 총 2개를 넘을 수 없고(구현한 학생이 적은 것부터 최대 2개) (2) 구현을 완료한 학생의 수가 50% 이내(소수점 첫째자리 반올림)여야 함. 단, (1), (2) 조건은 AND 조건임. 구현을 완료한 학생의 수가 50% 이내인 과제가 3개이고, 구현을 완료한 학생 수가 2번째와 3번째로 많은 과제를 구현한 학생의 수가 동일한 경우 가산점은 과제 출제 번호 뒷번호인 과제만 가산점 적용.
  - ✓ 가산점 부여 기준은 주어진 조건대로 모든 기능을 100% 구현을 완료(개요, 설계 등 보고서 내용 제외한 기능 구현만 확인)한 학생의 수(분반 통합 최종 성적 입력 전 출석부 상 학생 수 기준)가 가장 적은 과제부터 (주어진 조건대로 100% 구현하지 못한 학생 수) \* <3, 2> 임. 단, 구현을 완료한 학생의 수가 동일한 두 개 이상의 과제에 대해서는 낮은 배점을 곱함.
    - 예1. 최종 출석부 상 학생 수가 60명이고 3개 설계 과제 중, 1번 과제 10명 구현, 2번 과제 34명 구현, 3번 과제 20명 구현했을 경우, 가산점 부여 대상이 되는 과제는 50% 이내 학생이 수행(30명 이내)한 과제 1번과 3번이 되며, 가산점 부여 대상 과제는 1번과 2번이 됨. 1번 과제의 경우 (60-10)\*3=1000점, 3번 과제의 경우 (60-20)\*2=80점을 추가 부여. 2번의 경우 가산점 없음.
    - 예2. 최종 출석부 상 학생 수가 60명이고 3개 설계 과제 중, 1번 과제 20명 구현, 2번 과제 20명 구현, 3번 과제 20명 구현했을 경우, 가산점 부여 대상이 되는 과제는 50% 이내 학생이 수행(30명 이내)한 과제 1번, 2번, 3번이 되나, 가산점 부여 대상 과제는 2번 과제와 3번 과제가 대상이 됨. 2번 과제의 경우 (60-20)\*3=120점, 3번 과제의 경우 (60-20)\*2=80점 추가 부여. 1번 과제의 경우 가산점 없음.
  - ✓ 설계과제의 기준 및 각 과제별 "필수기능요건"은 각 설계 과제 명세서에 별도 설명. 설계과제명세서와 강의계획서 상 배점 기준이 다를 경우 해당 설계과제명세서의 배점 기준이 우선 적용
  - 과제마다 세부 기준은 과제 명세서에 명시되어 있음. 보고서에는 개요, 실행 결과 캡처 등, 소스코드(주석포함, 직접 구현하거나, 수정한 소스코드만 포함)이 포함되어야 함.
  - 보고서 5점, 실행 여부 95점(조건에 완전히 만족하는 경우 95점, 오류에 따라 감점)
  - 보고서 소스코드 포함하지 않았을 경우 해당 과제 배점에서 10% 감점 (설계 제출 방법 미준수 감점 10%를 더해 총 20% 감점)
  - ✓ 각 설계과제의 제출 유무는 점수로 판단하는 것이 아니라 주어진 과제의 세부 기준에 명시된 필수 기능 설계 및 구현으로 판단함. 예를 들어 특정 과제의 필수 기능 중 일부 기능만 구현했을 경우 해당 점수는 부여하나 과제는 미구현으로 판단하고 본 교과목 이수조건인 설계 과제 최소 구현 개수 2개에 포함시키지 않음
  - 예. 특정 과제의 필수 기능 중 일부 기능만 구현했을 경우 해당 점수는 부여하나 과제는 미구현으로 판단하고 본 교과목 이수조건인 설계 과제 최소 구현 개수 2개에 포함시키지 않음
- 출석 성적 배점 기준 (3점 만점)
  - ✓ **대면 수업 출석 점검은 첫 주부터 바로 실시하며, 출석 성적 배점에 포함** (단, 군 체대 등으로 인해 복학이 늦어진 경우에만 수업 첫 주 결석은 출석 성적 배점에 포함시키지 않음)
  - ✓ 학교(학사팀 등)에서 인정한 조기 취업 학생이 출석을 하지 못할 경우, 출석을 과제 수행으로 대체할 수 있기 때문에 출석 성적만 별도로 과제 수행을 통해 인정 가능함. 출석을 대체할 과제는 별도로 출제되며, 1회 결석 당 1개의 과제가 출제되고 1개 과제당 분량은 1시간 30분~2시간 정도 수행 가능한 정도가 될 것임
  - ✓ 따라서 조기 취업 학생도 본 평가 기준에 따라 성적을 부여 받기 때문에 설계 과제 수행은 물론이며, 중간/기말 시험을 반드시 치러야 함. 단, 조기 취업 학생의 경우 조기 취업이 확정되면 1주일 이내 조기 취업 사실을 교수에게 알려야 하며, 출석 대체 과제를 교수에게 별도로 확인해야 함. 1주일 이내 알리지 않을 경우 해당 주 차의 출석 대체 과제는 출제하지 않으며 결석으로 처리함
  - ✓ 개인적인 사정(친족 사망, 질병, 주요 면접 등)의 경우 증빙 서류 제출 시 출석 인정. 단, 질병으로 인한 입원의 경우 별도로 최대 3회까지 결석 허용하나 다른 사유의 결석을 포함 결석이 최대 8회 미만이어야 함
  - ✓ **usaint-)>전자출결/휴보강에서 출결을 확인하고 문제가 있을 경우 이메일 등으로 확인해야 함. 2주 이상이 지난 출결사항은 수정 불가능할 수 있음**
  - ✓ 1차 점검
    - 수업 시작 후 바로 출석 점검
    - 출석부(사진 출석부) 상 첫날은 1페이지부터 출석 확인, 둘째 날은 2페이지부터 확인
    - 1차 점검 시 출석이 확인되지 않은 경우 결석 처리.
    - 모든 학생의 출석을 부르기 전 입실 완료한 경우만 1차 점검 출석 인정
  - ✓ 2차 점검
    - 1차 점검 시 결석 처리된 학생을 대상으로 수업 시작 후 20~30분 이내 2차 출석 점검 후 지각 처리

- 2차 점검 시 출석이 확인되지 않은 경우 결석 처리
- ✓ 출석 성적 배점
  - 2회 결석 이내 3점, 2회 결석+1회 지각 2점, 2회 결석+2회 지각 1점, 3회 이상 결석 0점 부여
  - ✓ 지각 3번은 결석 1회와 동일하게 처리
- 수업 참여도 성적 배점 (2점 만점)
  - ✓ 강의 게시판 질문/대답
  - ✓ 강의 게시판 상 수업 내용 및 설계과제 관련 자발적인 학생들의 질문 및 답변의 횟수 와 실시간 화상 강의 시 질문 회수 등을 고려하여, 수업 내용 및 과제 관련 질문 5개 이상 2점, 3개 이상 1점, 기타 0점 부여 (기준 변경 가능)
  - ✓ 수업 내용 및 설계과제와 관련이 없는 질의 및 응답은 배점에서 제외 (예, 시험일정 관련 질문, 과제 제출 기한 질문 등 수업시간 및 게시판에 공지한 사항 관련 질문 등)
- 시험 및 과제 성적 확인
  - (가), (나) 분반 성적 평가는 통합 처리
  - 중간시험, 기말시험, 기본 과제, 설계 과제 성적은 추후 공지하며 충분히 검토할 수 있는 시간을 줌
  - 중간시험 채점 결과는 중간시험 치르고 채점 결과 복사본을 가능하면 2~3주 내에 학생들에게 배포하며 채점에 문제가 있을 경우 공정한 날짜에 한꺼번에 이의 신청(이메일, 전화, 방문만 가능)할 수 있음
  - 단, 이의 신청 기간 외 이의 신청은 불허함. 특히 기말시험을 치르고 최종 성적을 올리기가까지 주어진 시간이 많이 부족하고 바쁘기 때문에 이의 신청 기간이 아닐 때 이메일이나 전화로 이의 신청할 경우 최종 성적 감점 있음 (학생들의 이해 필요)
  - 기말시험 채점 결과는 구글클래스룸 게시판을 통해 공지한 날짜에 신청한 학생에게만 복사해서 배포하며 본인이 반드시 수령해야 함. **기말시험 복사 신청을 하고도 복사본을 기간 내에 수령하지 않을 경우 최종 성적 감점 있음**
  - 1번 설계과제 제출 시 학번/이름 옆에 비밀번호(6자리, 숫자 or 알파벳 조합, 대소문자 구분)를 기입해야 하며 비밀번호로 성적을 게시판에 공지함. 1번 설계과제 제출시 비밀번호를 제출하지 않는 학생은 학번으로 성적 공지. 자세한 비밀번호 관련 내용은 수업 시간에 제공지 예정. 단, 중간시험 비밀번호와 기말시험 비밀번호는 동일해야 함.
  - 이의 신청 기간 동안 여러 번 수정된 성적이 구글클래스룸 게시판에 공지
- 본 과목을 Pass하는(D학점 이상) 방법 (and 조건, 절대 변경 불가)
  - (1) 중간 또는 기말 시험을 모두 치르고 두 번의 시험 중 한번이라도 0점을 받지 않는다.
  - (2) 설계과제 4개 중 2개 이상을 완성한다. (설계 과제 중 최소 2개 과제는 각 과제별로 지정한 “필수구현” 부분을 구현한 완성으로 인정. 일부 기능의 구현의 경우 최종 성적에는 반영되는 설계 과제는 미완성으로 처리. 단, 일부 기능 및 출력의 사소한 에러는 허용 가능)
  - (3) 결석은 8회(일) 미만으로 한다.
  - (4) 남에게 과제 결과물을 일부 및 전부를 절대로 보여주지 않는다. (남의 과제 결과물 및 인터넷에 게시된 내용을 일부 및 전부를 복사한 경우, 과제 마감 전 인터넷 등에 공개한 경우 및 설계한 내용이 공유되어 유사하게 구현되었을 경우 등 기타 부정행위라고 간주되는 모든 행위 포함. 이 경우 수업 도중에 알리지 않고 최종 성적 공지 시 알릴 예정임). \*\*\* git 등에 자기가 수행 과제 소스코드 등을 올리고 싶을 땐 본 수업을 마친 후(12월 말 이후)에 올리기 바람. 과제로 출제된 명세와 공개한 소스코드는 git에 올리는 것은 절대 불가하며, 자기가 직접 작성한 코드와 보고서만 올려야 함.
  - (5) 인터넷 사이트 등에서 과제 대행을 부탁하지 않는다.
- 기타
  - 긴급한 공지는 구글클래스룸 게시판을 통해 진행. 수업시간 전후 반드시 확인해야 함
  - 학생들은 게시판에 수업 및 과제 관련 질문을 해야 하며, 과제 관련해서 궁금한 것은 게시판, 직접 연구실 방문이 가능함. 단, 연구실 방문은 조교에게 미리 메일로 연락하여 방문 예약을 해야 함. 질문은 게시판을 통해서만 가능
  - 수업 내용 및 과제 내용 관련 질문은 어떠한 것도 수용 가능하나, 직접 답을 알려주지는 않음. 과제를 할 수 있는 방법을 설명하는 수준이 될 것임
  - 과제 관련 질문이 많으면 많을수록 더 많은 힌트를 줄 수 있으니, 주저하지 말고 질문하기 바람
  - 학생들의 수업 참여 정도와 수업 시간 태도에 따라 상세 평가 방법은 변동 가능함
  - Oracle VM VirtualBox 등 가상머신 -> 리눅스 우분투 배포판 버전 설치
  - gdb manual
    - ✓ <https://sourceware.org/gdb/current/onlinedocs/gdb/>
  - Guide to using gdb to debug child process
    - ✓ <https://moss.cs.iit.edu/cs351/gdb-inferiors.html>
  - Vim manual

- 
- ✓ <http://vimdoc.sourceforge.net/html/doc/help.html>
  - git
  - ✓ <http://www-cs-students.stanford.edu/~blynn/gitmagic/>
  - xv6
  - ✓ <https://pdos.csail.mit.edu/6.S081/2020/xv6/book-riscv-rev1.pdf>
  - ✓ <https://pdos.csail.mit.edu/6.828/2022/xv6/book-riscv-rev3.pdf>
  - ✓ <https://github.com/palladian1/xv6-annotated>

## 6. 강의 일정 (주차별 강의 내용은 학생들 수준에 따라 변경 가능)

주	일/요일	강의 내용	3 Easy Pieces 교재	Silberschatz 교재	Stalling 교재	비고
1.0	09월 04일 (월)	강의 및 과제 출제 계획				과제1 출제
	09월 06일 (수)	가상화 - 프로세서 가상화	2장	2장	2장	
2.0	09월 11일 (월)	가상화 - 프로세서 가상화	4장	3장	3장	
	09월 13일 (수)	가상화 - 프로세서 가상화	5장, 6장	3장	3장	
3.0	09월 18일 (월)	스케줄링	7장	3장	3장	과제1 마감/과제 2출제
	09월 20일 (수)	스케줄링	8장	5장	9장, 10장	
4.0	09월 25일 (월)	가상화 - 물리메모리	20장	9장, 10장	7장, 8장	
	09월 27일 (수)	가상화 - 물리메모리	21장	9장, 10장	7장, 8장	사전녹화온라인(추석연휴전날)
5.0	10월 02일 (월)	가상화 - 메모리 (메모리 API)	21장	9장, 10장	7장, 8장	사전녹화온라인(추석연휴다음날)
	10월 04일 (수)	가상화 - 메모리 (주소 변환)	13장	9장, 10장	7장, 8장	
6.0	10월 09일 (월)	가상화 - 메모리 (세그멘테이션)	15장	9장, 10장	7장, 8장	사전녹화온라인(한글날)
	10월 11일 (수)	가상화 - 메모리 (자유공간관리)	16장	9장, 10장	7장, 8장	
7.0	10월 16일 (월)	가상화 - 메모리 (페이징)	17장	9장, 10장	7장, 8장	과제2 마감 / 과제3 출제
	10월 18일 (수)	가상화 - 메모리 (페이징)	17장	9장, 10장	7장, 8장	
8.5	10월 23일 (월)	병행성 - 쓰레드	19장	9장, 10장	7장, 8장	사전녹화온라인(출장)
	10월 25일 (수)	병행성 - 쓰레드	19장	9장, 10장	7장, 8장	사전녹화온라인(출장)
	10월 27일 (금)	중간시험				17:00~20:00
9.5	10월 30일 (월)	병행성 - 쓰레드	26장	4장	4장	
	11월 01일 (수)	병행성 - 락	28장	6장	5장	
10.5	11월 06일 (월)	병행성 - 조건변수	29장	6장, 8장	5장, 6장	
	11월 08일 (수)	병행성 - 세마포어	30장	6장	5장	
11.5	11월 13일 (월)	병행성 - 일반적인 문제	31장	6장	5장	과제3 마감 / 과제4 출제
	11월 15일 (수)	지속성(객체저장) - HDD	32장	7장	5장	
12.5	11월 20일 (월)	지속성(객체저장) - RAID	37장	11장	11장	
	11월 22일 (수)	지속성(객체저장) - 파일시스템	38장	11장	12장	
13.5	11월 27일 (월)	지속성(객체저장) - crash	40장	13장, 15장	12장	
	11월 29일 (수)	지속성(객체저장) - LSF	44장	14장, 15장	12장	
15.0	12월 04일 (월)	정리				과제 4 마감
	12월 06일 (수)	기말시험				17:00~20:00
	12월 11일 (월)	예비				

※ 일정 별 강의 내용 및 과제 출제 및 마감일은 변경 가능.

※ 과제 마감일은 아래 7번 설계과제 설명에 나와 있는 마감일에 따름. 만약 강의계획서와 과제명세서 내에 있는 제출 마감일이 다를 경우 과제 명세서에 명시된 제출 마감일에 따름.

※ 중간고사 및 기말고사는 12월 11일 및 13일 보강형태로 이루어짐.

## 7. 설계과제

번호	과제 제목	과제 출제	과제 마감	난이도
1	xv6 설치	09월 04일(월)	09월 17일(일)	하
2	xv6 (Linux/BSD) OS 시스템콜 추가	09월 17일(일)	10월 15일(일)	하
3	xv6 스케줄러 구현 / Linux vs BSD 스케줄러 분석	10월 15일(일)	11월 12일(일)	중
4	xv6 메모리관리 및 파일시스템 구현	11월 12일(일)	12월 03일(일)	상