

Operating Systems Project 03

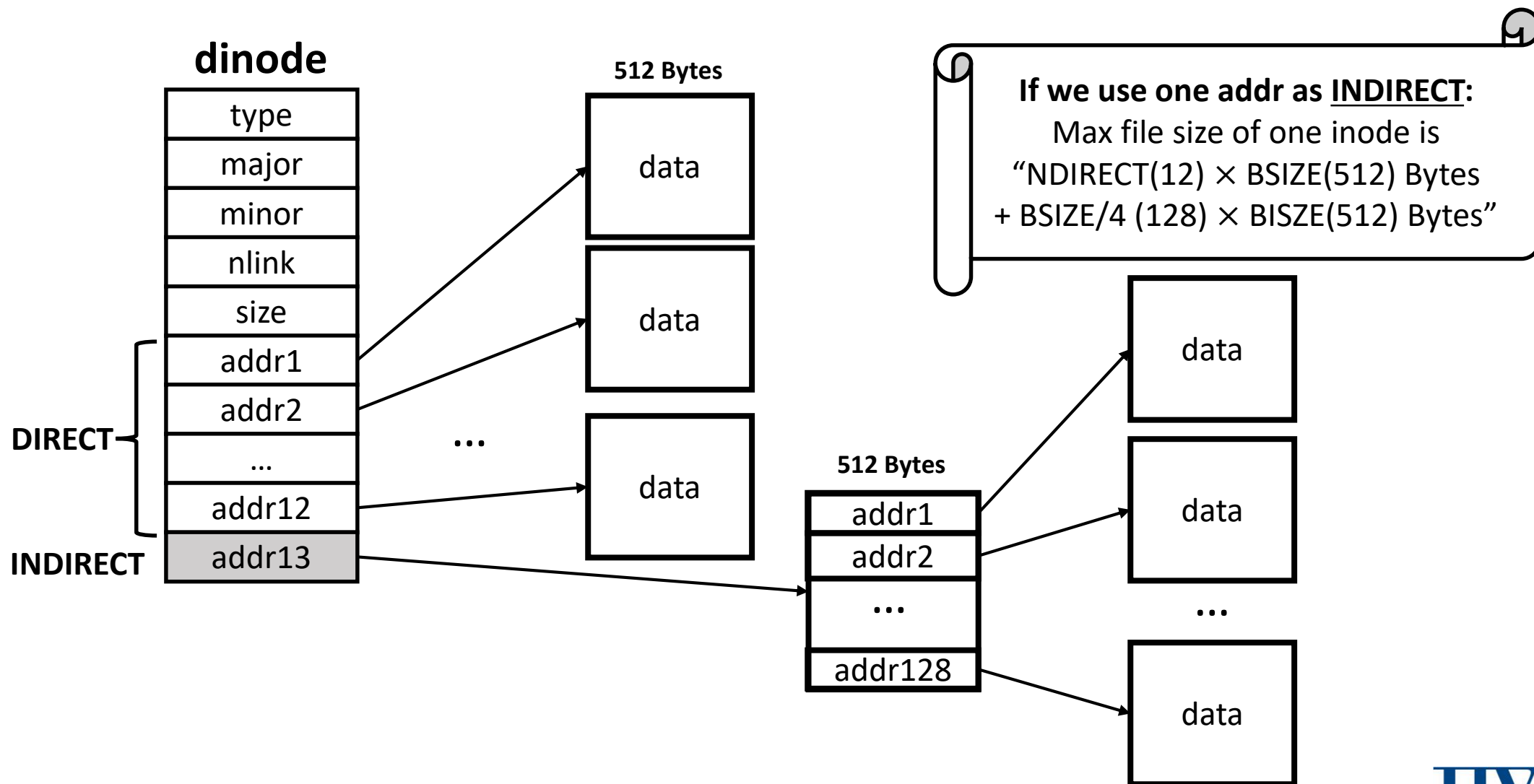
- Implementing double indirect in xv6-

Due date : 11:59 pm June 23

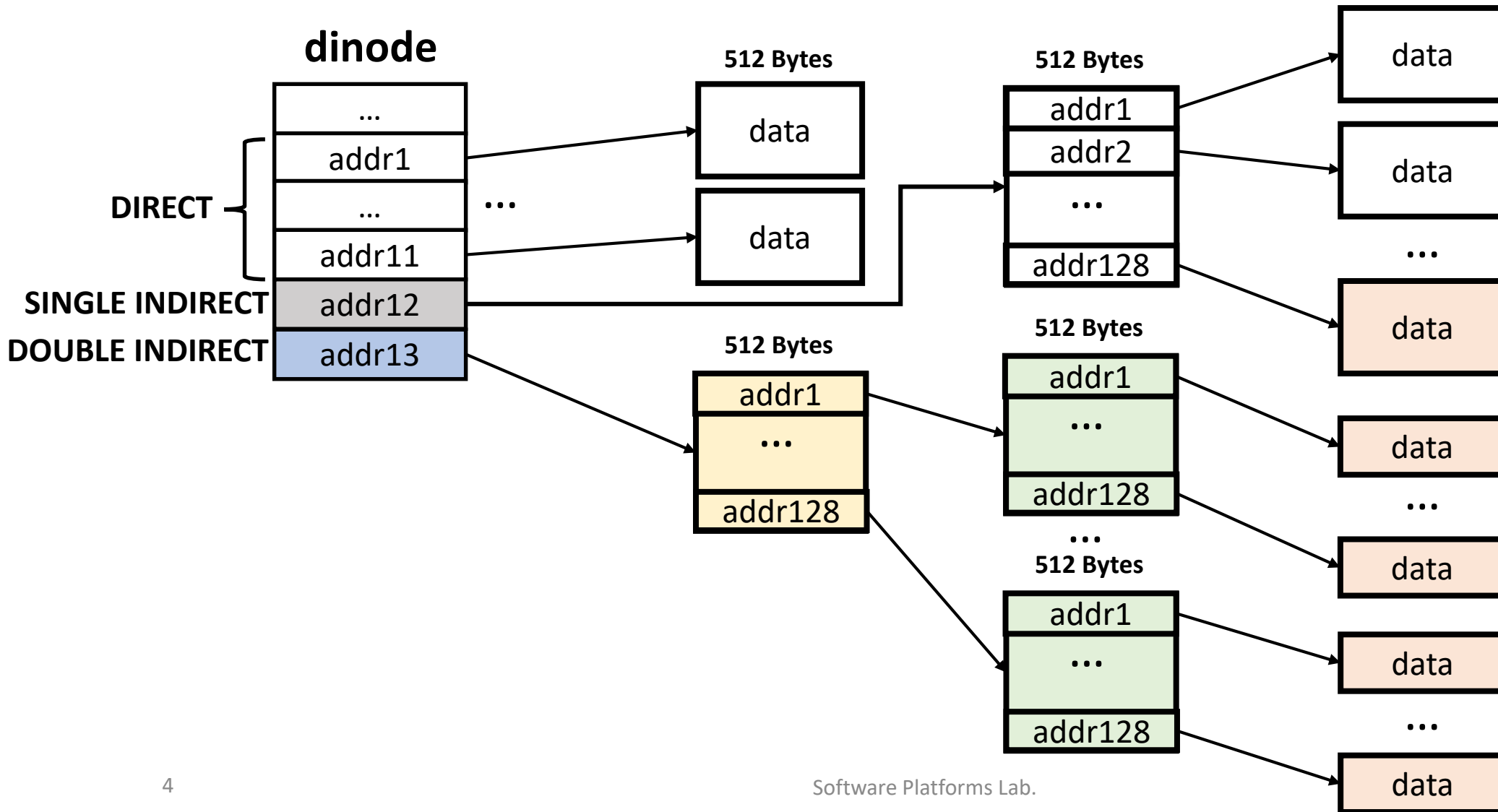
Double indirect

- 실습 시간에 살펴보았듯이, xv6의 inode에서는 큰 파일을 다루기 위해 한 번의 indirect를 지원합니다.
- 하지만 한 번의 indirect만으로는 최대 140개 블록에 해당하는 크기밖에는 파일을 쓸 수 없습니다.
- 이번 과제의 목표는 xv6가 더 큰 파일을 다루게 할 수 있게 하기 위해, inode에 double indirect를 구현하는 것입니다.

xv6's original inode content



After implementing double indirect



힌트

- 수정해야 할 함수는 두 가지입니다.
 - bmap
 - itrunc
- fs.h, param.h에 정의된 매크로 상수들의 값들을 일부 변경해야 합니다.
 - FSSIZE는 20000 이상의 값을 사용하도록 합니다.
 - 그 외에도 필요한 것이 있다면 변경하거나 추가해도 됩니다.
- 변경된 사항이 inode와 dinode 구조체에도 올바르게 반영되어야 합니다.

테스트 프로그램

- 업로드 된 file_test.c를 같이 빌드하여 간단한 테스트를 해볼 수 있습니다.
- Test 1~3으로 이루어져 있으며, test 1은 파일에 8MB 쓰기, test 2는 8MB를 파일에서 읽기, test 3는 test 1과 2를 10회 반복하며 문제가 발생하지 않는지 확인합니다.
- xv6의 디자인상 전체 테스트에 상당한 시간이 소요될 수 있습니다. 각 테스트 결과가 나오기를 오래 기다려 보시기를 권장합니다.

평가

- **Completeness**: 명세의 요구조건에 맞게 xv6가 올바르게 동작해야 합니다.
- **Defensiveness**: 발생할 수 있는 예외 상황에 대처할 수 있어야 합니다.
- **Wiki&Comment**: 테스트 프로그램과 위키를 기준으로 채점이 진행되므로 위키는 최대한 상세히 작성되어야 합니다.
- **Deadline**: 데드라인을 반드시 지켜야 하며, 데드라인 이전 마지막으로 commit/push된 코드를 기준으로 채점됩니다.
- **Do NOT copy or share your code.**

평가지표

평가 기준은 다음과 같으며, 각 명세의 일부분을 완성하였다면 부분점수가 주어집니다.
실패하는 테스트가 있을 시 해당 부분에서 감점이 이루어집니다.

평가 기준	배점
큰 파일 쓰기	30
큰 파일 읽기	30
파일 삭제 및 재생성	20
Wiki	20
Total	100