Project #3

Dept. of Computer Science

Hanyang University

File system

Goal of project

- xv6의 파일시스템은 최소한의 기능으로 구현되어 있습니다.
- xv6에서는 Multi Indirect, Symbolic Link 그리고 Buffered I/O 기능 등이 구현되어 있지 않습니다.
- 위 기능들을 구현하여 xv6를 개선하는 것을 목표로 합니다.

Multi Indirect

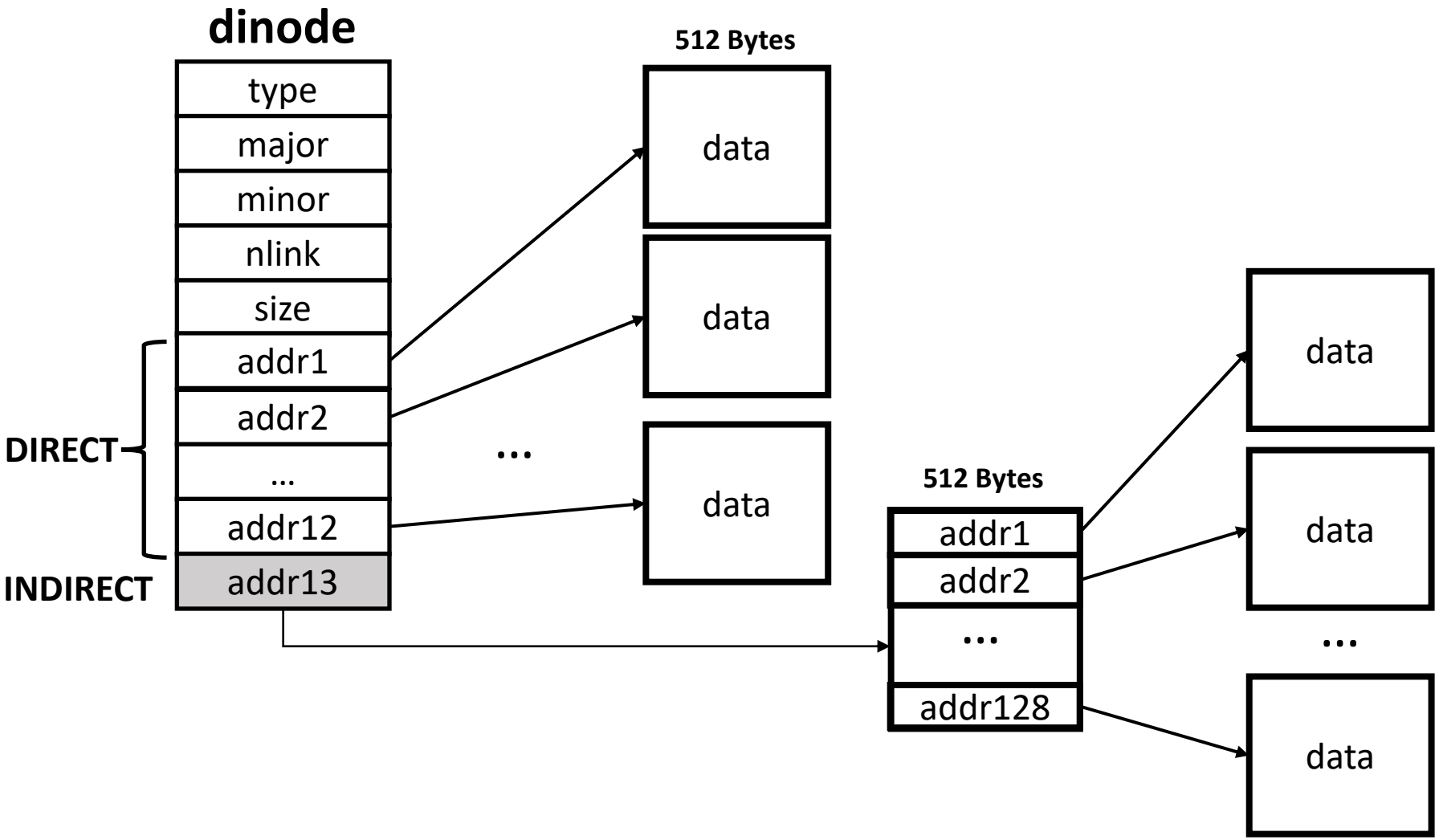
Multi Indirect

- **xv6는 Direct와 Single indirect를 통해 파일의 정보를 저장합니다.**

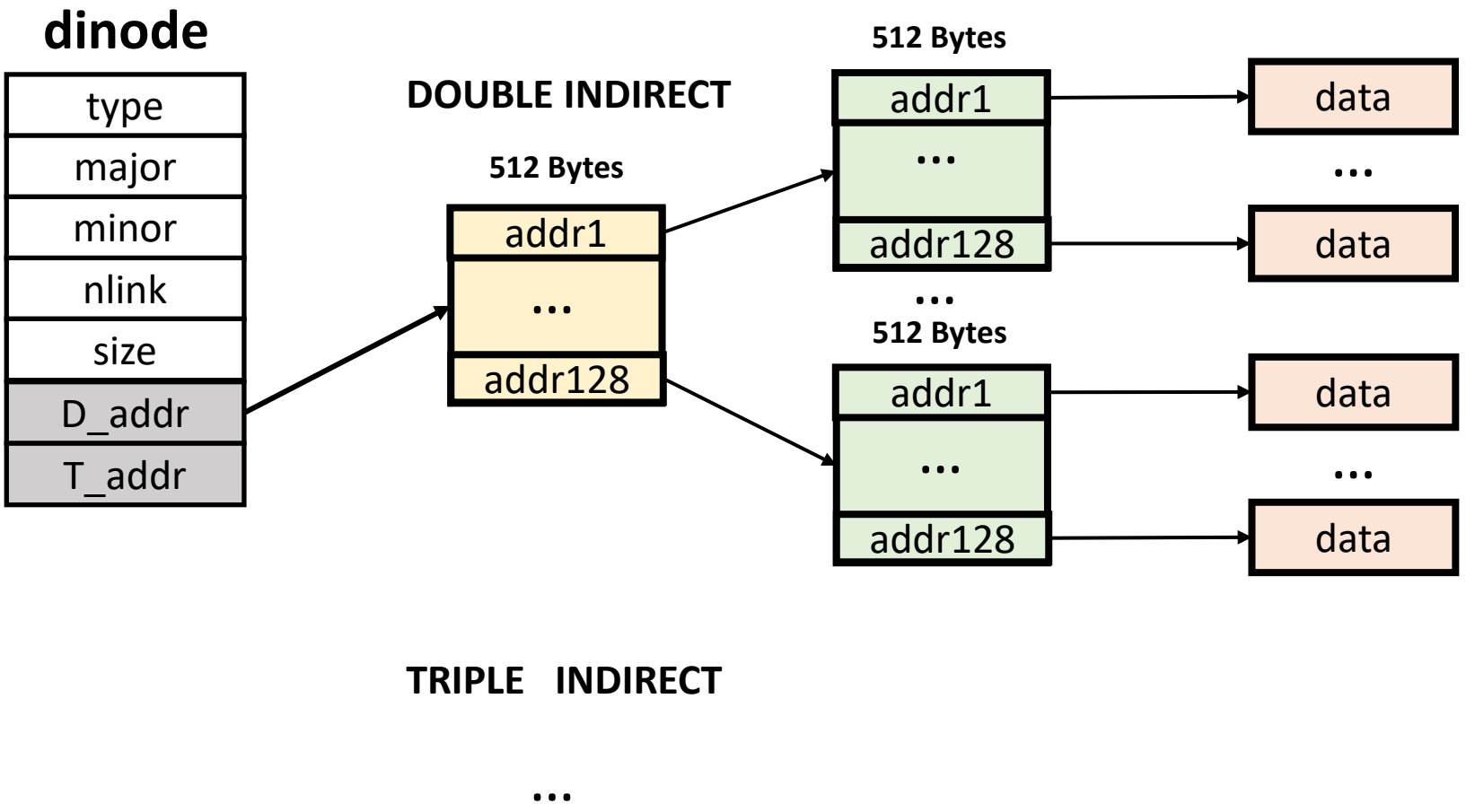
- **기존 방식은 최대 140개 블록에 해당하는 파일만 쓸 수 있다는 한계가 있습니다. (512B * 140 = 70KB)**

- **이번 과제는 xv6가 더 큰 파일을 다루게 할 수 있게 하기 위해, multi indirect를 구현하는 것을 목표로 합니다.**

Multi Indirect



Multi Indirect



Multi Indirect

- Double Indirect는 최대 16384개 블록 약 8MB의 파일을 관리할 수 있습니다.
- Triple Indirect는 약 1GB까지 파일을 관리할 수 있습니다.
- 기본적인 xv6의 File System은 최대 데이터 블록 개수에 제한이 있습니다.
  “FSSIZE” Parameter를 조절 하여 이 제한을 해제할 수 있습니다.

Symbolic Link

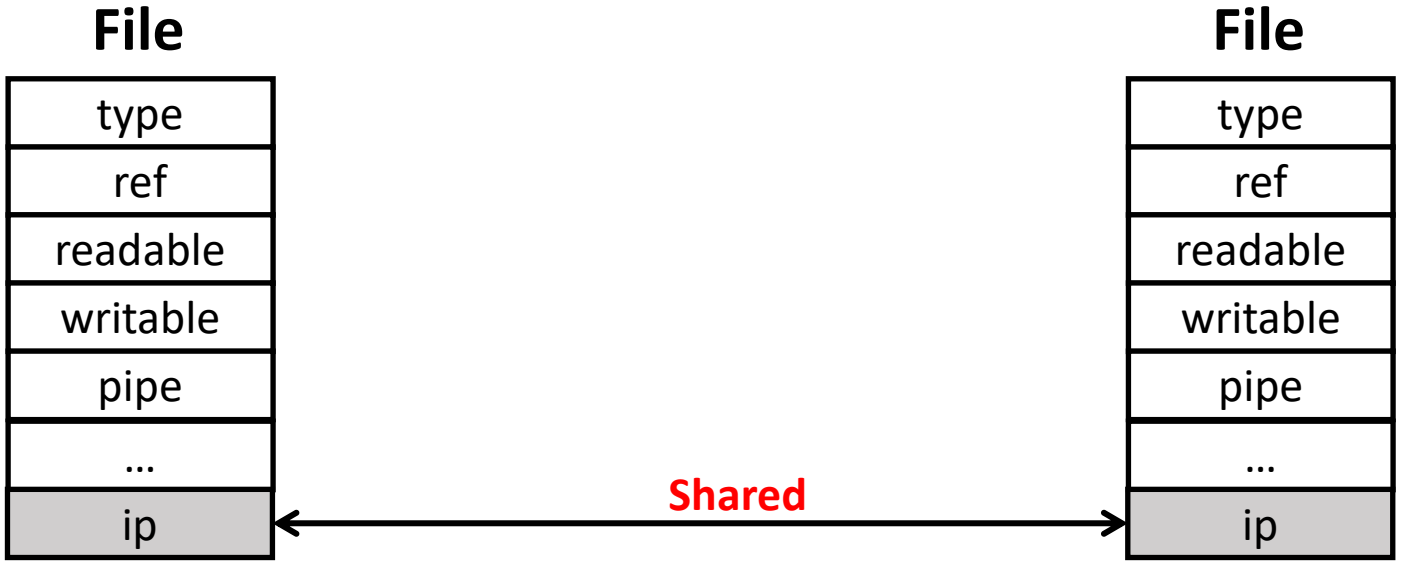
Symbolic Link

- **xv6는 Hard Link를 지원하지만, Symbolic Link를 지원하지 않습니다.**

- **Symbolic Link를 xv6에 구현하는 것을 목표로 합니다.**

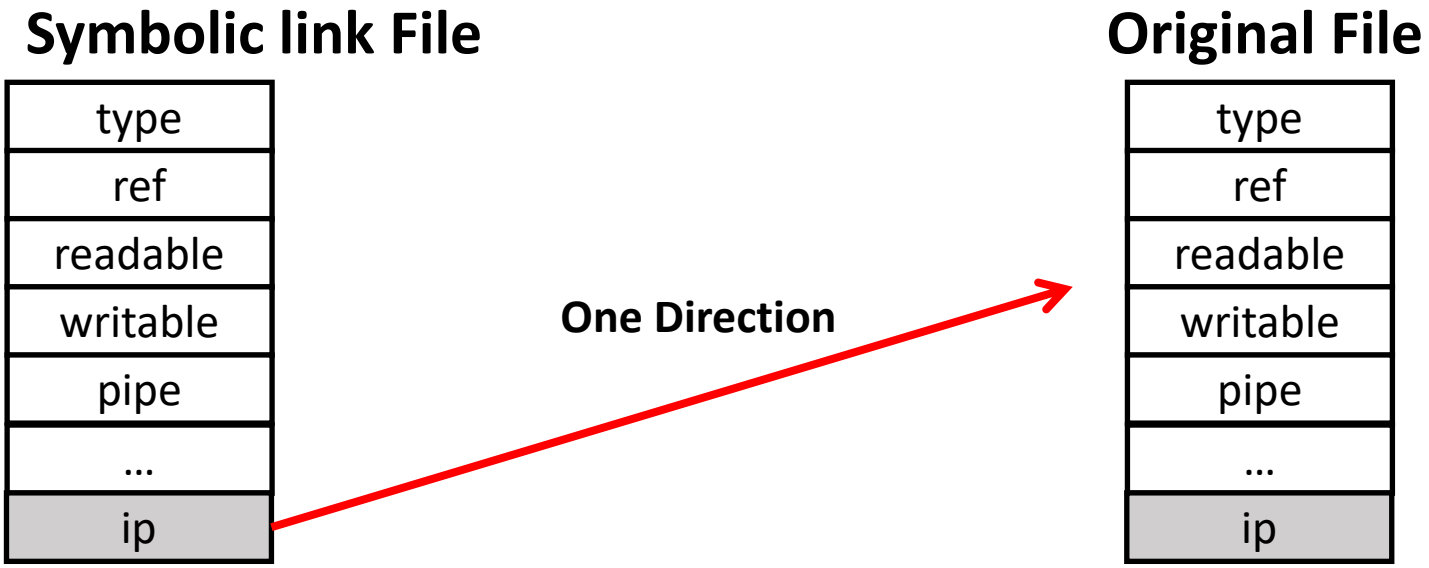
Symbolic Link

- **Hard Link 파일은 원본 파일의 I-node를 공유하는 파일입니다.**
- **각각의 파일은 별도의 파일이지만, 서로의 변경 내역은 공유됩니다.**
- **한 파일이 제거되어도 다른 파일에 영향을 주지 않아야 합니다.**



Symbolic Link

- Symbolic Link는 한 파일이 다른 파일을 가리키는 파일입니다.(바로가기)
- Process가 Symbolic Link 파일에 접근할 경우, Original File로 Redirection 되어야 합니다.
- 원본 파일이 제거될 경우 Symbolic Link File은 남아있지만, 데이터에 접근할 수 없어야 합니다.



Symbolic Link - Specification

- 기존 xv6에 존재하는 ln user program을 변경하여,
  hard link와 Symbolic link를 만들 수 있도록 변경합니다.

- Shell에 ln [–h] [old] [new]와 같이 명령어를 입력할 경우
  new 파일을 old 파일의 Hard 링크 파일로 만들어야 합니다.

- Shell에 ln [–s] [old] [new] 와 같이 명령어를 입력할 경우
  new 파일을 old 파일의 Symbolic 링크 파일로 만들어야 합니다.

- Hard, Symbolic 링크 파일은 각각 기존의 write, read, open, close 등과 같은
  File Operation을 적절하게 진행할 수 있어야 합니다.

- 구현 과정에서 systemcall이 필요할 경우 적절히 만드시면 됩니다.

Sync

Sync

- 일반적으로, Disk I/O는 컴퓨터 Operation중에서 가장 느린 Operation입니다.

- 기존 xv6는 write operation에 대하여, group flush를 진행합니다.
  해당 방식은 특정 프로세스가 다수의 Write Operation을 발생시킬 때 성능을 저하시키는 문제점을 가지고 있습니다.

- 위의 문제를 해결하기 위해,
  sync 함수가 호출될 때만 flush하도록 하는 Buffered I/O를 구현합니다.

Sync - Specification

- 기존의 begin_op, end_op, commit 등의 함수를 수정하여 Group Flush가 아닌, Buffered I/O가 구현되어야 합니다.
- 더 이상 Group Commit은 발생하지 않습니다.
- int sync(void)시스템 콜을 구현하여야 합니다.
  성공 시 flush된 block수가, 실패 시 -1을 반환해야 합니다.
- Sync 호출 시, 현재 Buffer에 있는 모든 Dirty Buffer가 Flush되어야합니다.
- Buffer에 공간이 부족한 경우, 강제적으로 sync를 발생시켜야 합니다.
- Sync를 호출하지 않고, file이 close될 경우, 변경 내용이 반영되지 않아야 합니다.

Sync - Specification

- 기존 BufferSize는 10 * 3* 512B =15K로 상당히 작습니다. 채점시 Buffer의 크기는 기존보다 커질 수 있습니다.

- 과제 구현시, Parameter 값들을 변경하고 싶으시다면 param.h파일을 참조하시기 바랍니다.

Tips & Evaluation

Tips

- 기본 xv6를 기반으로 동작하면 됩니다.
  - 과제에 제시된 3가지 항목은 각각 다른 항목에서도 정상적으로 동작하여야 합니다.
  - 각각의 기능이 유기적으로 동작하지 않는다면, 감점 사항으로 작용할 수 있습니다.

- 어떤 부분을 어떻게 고쳐야 할지 막막하다면, xv6 코드를 참조하시기 바랍니다.
  - Hard Link와 Symbolic Link의 개념에 유의하시어 구현하면, 쉽게 구현할 수 있습니다.
  - 기존 Link 함수를 참고하시면 과제 진행에 도움을 얻으실 수 있습니다.

- Sync 구현 시, begin_op와 end_op 함수의 동작을 참고하시기 바랍니다.

- 과제 진행 중 어렵다면, xv6의 파일시스템 구조분석을 권장 드립니다.

- GitLab에 새로운 디렉토리를 생성하셔도 괜찮습니다. (ex. project03)
  - 기존 Project1,2내용을 별도 디렉토리에 저장하셔도 괜찮습니다. (ex. Project01, Project02)

Evaluation

- **Code**
  - 명세의 요구 조건을 모두 올바르게 구현해야 합니다.
  - 코드에는 주석이 함께 작성되어야 합니다.

- **Wiki**
  - 테스트 프로그램과 위키를 기준으로 채점됩니다.
  - 위키는 실제로 동작하는 장면과 함께 본인의 디자인에 대해 상세히 작성되어야 합니다.

- **Submission**
  - 데드라인을 반드시 지켜야 합니다.
  - 데드라인 이전에 GitLab에 마지막으로 commit/push 된 코드를 기준으로 채점됩니다.

Code

- **Completeness**
  - 명세의 요구 조건을 모두 올바르게 구현해야 합니다.
- **Defensiveness**
  - 발생할 수 있는 예외 상황에 대처할 수 있어야 합니다.
- **Comment**
  - 코드에는 반드시 주석이 있어야 합니다.
- **DO NOT ASK OR SHARE CODE !!**

Wiki

- **Design**
  - 명세에서 요구하는 조건에 대해서 **어떻게 구현할 계획**인지, 어떤 자료구조와 알고리즘이 필요한지, **자신만의 디자인**을 서술합니다.
- **Implement**
  - 본인이 **새롭게 구현하거나 수정한 부분**에 대해서 무엇이 기존과 어떻게 다른지, 해당 코드가 무엇을 목적으로 하는지에 대한 설명을 구체적으로 서술합니다.
- **Result**
  - 컴파일 및 **실행 과정**과, 해당 명세에서 요구한 부분이 정상적으로 동작하는 **실행 결과**를 첨부하고, 이에 대한 동작 과정에 대해 설명합니다.
- **Trouble shooting**
  - 과제를 수행하면서 **마주하였던 문제**와 이에 대한 **해결 과정**을 서술합니다. 혹여 문제를 해결하지 못하였다면 어떤 문제였고 어떻게 해결하려 하였는지에 대해서 서술합니다 .
- **And whatever you want ☺**

Submission

- You should upload your code to HYU GitLab repository.
- You should upload your wiki to HYU LMS (only .pdf)
- Wiki file name is “ELE3021_ project03_[classNO]_ [studentID].pdf”
- **Due date : 2023. 06. 16. 23:59 (No late submission allowed)**

Thank you :)