

## 기초 프로젝트 최종 보고서 ( II 단계)

프로젝트 분야/주제	문서공유 프로그램					총 투입인원	2명
프로젝트 명	국문	문서공유 프로그램					
	영문	Document sharing program					
수행기간	2016. 11. 13~ 2016. 12. 06						
참여자	소속	참여자성명	학년	학번	휴대전화	이메일	
	선문대학교	김경보	2	2012244048	010-3567-2693	boriskim936@naver.com	
	선문대학교	김의섭	3	2012244030	01094942318	dmltjq2524@gmail.com	
	-	-	-	-	-	-	
	-	-	-	-	-	-	

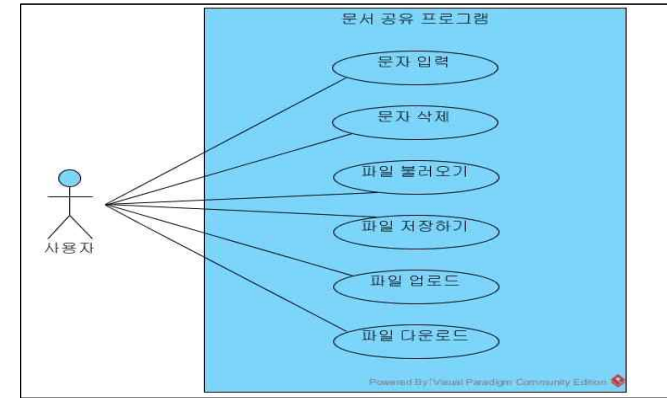
## 개발결과

## 1. 개발 결과물에 대한 요약

- 4중 연결 리스트를 이용하여 구현한 메모장에 서버·클라이언트의 개념을 더해 문서를 공유할 수 있도록 해 주는 프로그램
- 클라이언트(메모장)는 자신이 작성한 파일을 문자 하나하나 씩 한 노드에 저장 하여 해당 노드를 4중으로 연결해 문서를 구성
- 클라이언트는 서버에게 다운로드·업로드 요청이 가능
- 업로드 요청 시 클라이언트는 글쓴이, 제목 과 자신의 리스트에 모든 노드 합쳐 하나의 문자열로 서버에 전송
- 서버는 업로드 요청 메시지를 수신 하면 해당 문자열을 분리해 글쓴이, 제목, 내용의 정보를 새로운 노드에 저장하며 해당 노드를 자신의 연결리스트에 저장
- 다운로드 요청 시 클라이언트는 서버로부터 파일 목록을 수신해 화면에 표시하며 다운받은 파일의 제목을 서버로 전송
- 서버는 수신한 파일명에 해당하는 파일을 자신의 리스트에서 검색하며 검색결과를 클라이언트로 전송
- 클라이언트는 수신한 문자열(파일의 내용)을 한 문자 씩 자신의 리스트에 저장(기존에 입력했던 리스트는 초기화함)

## 2. 요구사항 분석 대비 핵심 개발 사항

## 2-1 Usecase Digram



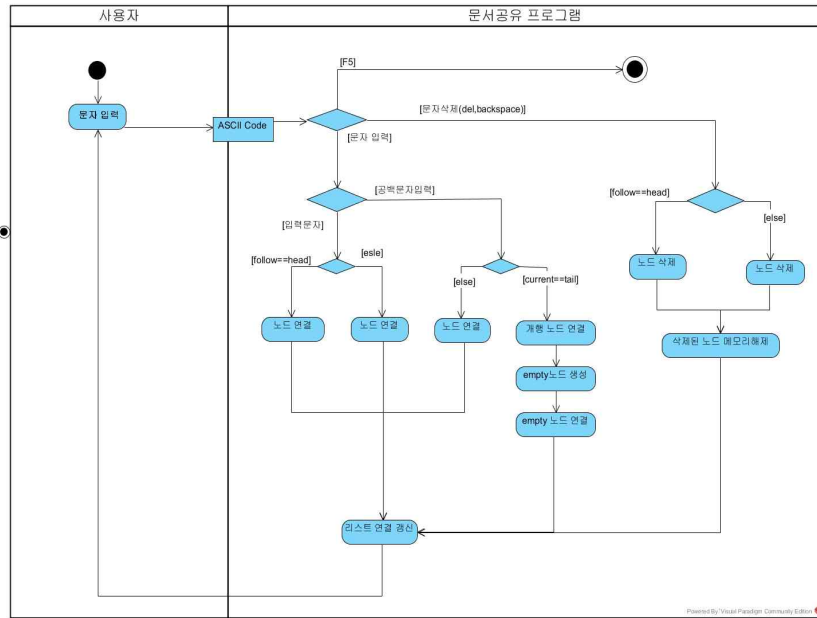
## 2-1 Usecase Discription

대분류	구분	구분		아이디	내용
		중분류	소분류		
문자	문자	문자 입력	-	KK_STR_1	<ul style="list-style-type: none"> <li>KK_FCS_01을 통해 얻은 포커스 뒤에 노드 삽입</li> <li>노드 연결 시 각각의 노드는 상하좌우를 표현하는 4개의 포인터를 가지고 있는 4중 연결리스트를 사용하며 해당 포인터를 이용해 문자 사이사이를 자유롭게 이동</li> <li>노드 연결 시 기존의 연결 과 새로 연결되는 노드를 고려하여 리스트간의 연결 갱신</li> </ul>
		문자 삭제	-	KK_STR_2	<ul style="list-style-type: none"> <li>KK_FCS_01을 통해 얻은 포커스에 노드 삭제</li> <li>노드 삭제 시 기존의 연결 과 새로 연결되는 노드를 고려하여 리스트간의 연결 갱신</li> <li>삭제된 노드는 메모리해제</li> </ul>
파일	파일	파일 불러오기	-	KK_FILE_1	<ul style="list-style-type: none"> <li>파일을 읽어 문자 단위로 노드에 저장하여 리스트에 연결</li> </ul>
		파일 저장하기	-	KK_FILE_2	<ul style="list-style-type: none"> <li>리스트에 연결되어 있는 노드들을 파일로 출력</li> <li>파일 저장 시 저장할 파일의 이름을 입력 받고 해당 이름의 txt확장자로 파일을 저장</li> </ul>
	파일	파일업로드	서버	KK_FILE_3_S	<ul style="list-style-type: none"> <li>파일 업로드 요청 수신 시 서버는 해당 클라이언트를 위한 비동기 쓰레드를 생성하고 해당 클라이언트의 연결요청을 승인</li> <li>파일 업로드 요청 시 업로드 할 파일을 저장, 제목, 내용을 포함하는 file구조체에 저장</li> <li>file 구조체(공유자원)에 접근하기 위하여 뮤텍스를 사용하여 상호배제, 진행조건 위배 등의 공유자원 문제를 해결</li> </ul>
			클라이언트	KK_FILE_3_C	<ul style="list-style-type: none"> <li>파일 업로드 시 클라이언트는 서버의 IP, PORT를 입력하여 연결 요청을 하며 연결이 승인되면 업로드 작업 시작</li> <li>파일 업로드 시 클라이언트는 자신의 리스트의 모든 노드(문자)를 하나의 문자열로 만들어 서버로 전송</li> </ul>
		파일다운로드	서버	KK_FILE_4_S	<ul style="list-style-type: none"> <li>파일 다운로드 요청 시 서버는 해당 클라이언트를 위한 비동기 쓰레드를 생성하고 해당 클라이언트의 연결요청을 승인</li> <li>서버는 클라이언트에게 자신의 파일 목록을 전송</li> <li>클라이언트로부터 파일의 제목을 수신하면 해당 파일의 내용을 클라이언트에게 전송</li> <li>file 구조체(공유자원)에 접근하기 위하여 뮤텍스를 사용하여 상호배제, 진행조건 위배 등의 공유자원 문제를 해결</li> </ul>
			클라이언트	KK_FILE_4_C	<ul style="list-style-type: none"> <li>파일 다운로드 시 클라이언트는 서버의 IP, PORT를 입력하여 연결 요청</li> <li>연결요청 승인 시 클라이언트는 서버로부터 파일 목록을 수신</li> <li>클라이언트는 자신이 다운받고자 하는 파일의 이름을 입력하여 서버로 전송</li> <li>해당 파일의 내용을 수신하기 위해 클라이언트의 기존 리스트는 비우고 서버로부터 수신한 문자열을 한 문자 씩 읽어, 한 문자씩 리스트에 연결</li> </ul>
포커스	포커스 추적	-	-	KK_FCS_1	<ul style="list-style-type: none"> <li>입력 위치 및 삭제 위치 등 현재 초점이 맞춰있는 노드를 찾을</li> </ul>

## 상세개발내용

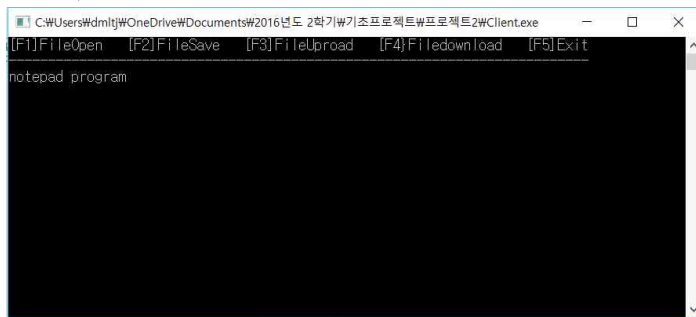
## 1. 상세 개발 내용

## 1-1. 문자 삽입 &amp; 삭제

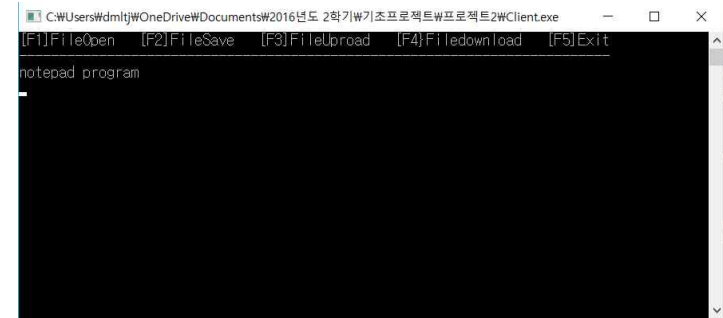


## a. 문자 삽입

- 문자 입력 시 해당 문자의 ASCII코드 값 으로 저장
- 입력문자와 개행 문자를 구별하여 문자 삽입 진행(현 포커스(current)뒤에 삽입)
- 노드 삽입 시 현 포커스의 위치(current) 및 포커스의 up, down 포인터 값에 따라 개별적인 노드삽입 루틴을 실행하여 기존노드와 새로운 노드 간 연결
- 개행문자 삽입 시 종류에 상관없이 모두 10 대입 (키보드의 enter와 'n'문자의 ASCII값은 각각 13, 10 이므로 10으로 통일하여 저장)
- 실행화면



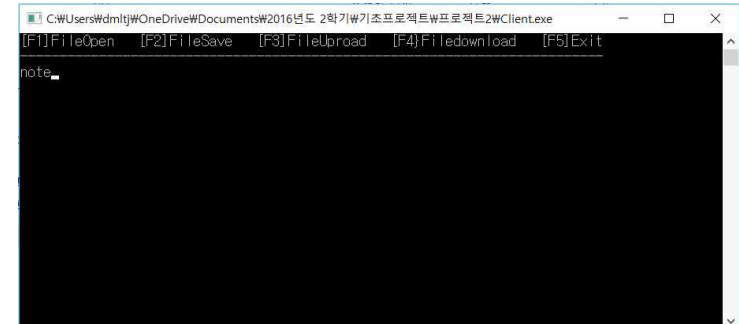
• notepad라는 문자열 입력



- notepad program 입력 후 개행 문자 입력
- 개행 문자 입력 시 빈 공간을 표현하기 위해 empty노드 활용

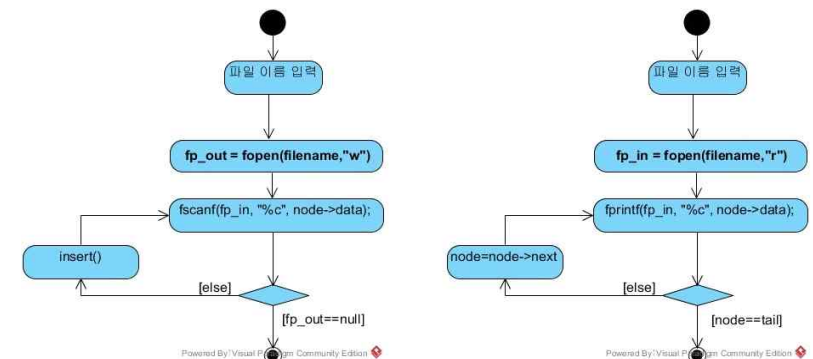
## b. 문자 삭제

- BackSpace 기능 구현(del 미구현)
- BackSpace 입력 시 현재 포커스의 앞 노드 삭제
- 노드 삭제 시 현 포커스의 위치 및 up, down 포인터 값에 따라 개별적인 노드삭제 루틴을 실행하여 기존노드와의 연결 갱신
- 삭제된 노드는 메모리 해제
- 실행화면



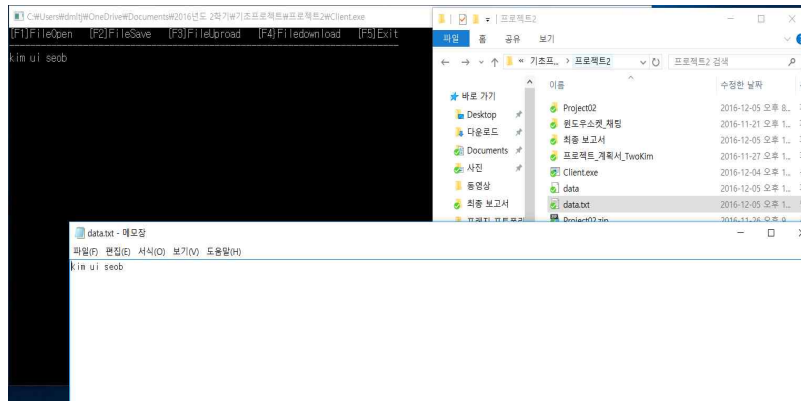
- notepad 입력 후 BackSpace 3번 입력

## 1-2. 파일 저장 &amp; 불러오기



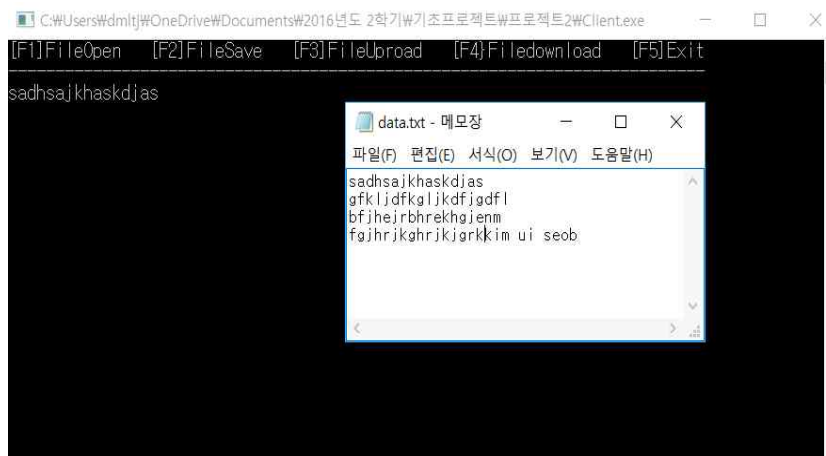
## a. 파일 저장하기

- 리스트의 head~tail까지 모든 노드의 문자를 텍스트 파일로 저장
- 실행화면



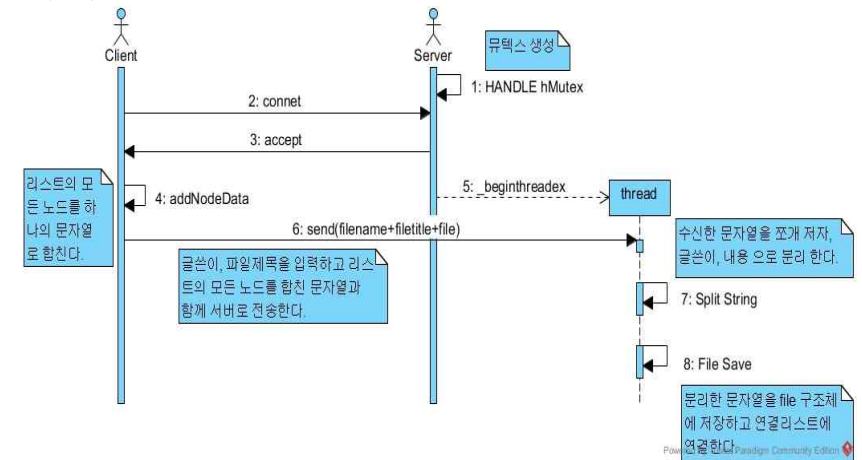
## b. 파일 불러오기

- 파일의 EOF까지 모든 문자를 읽어 문자 하나하나 리스트에 저장
- 실행화면



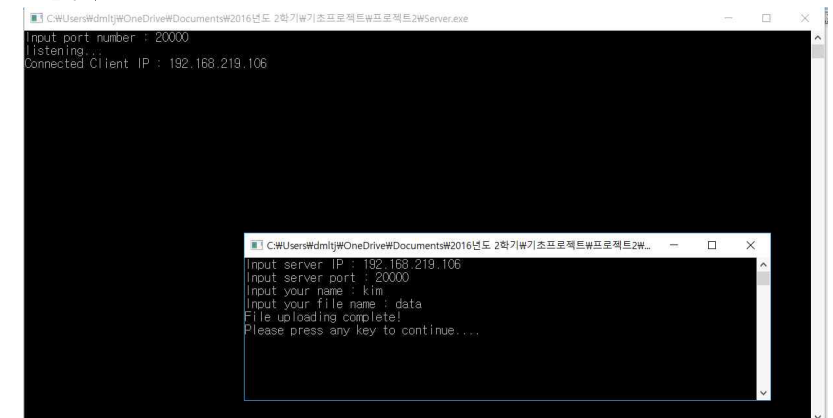
## 1-3. 파일 업로드 &amp; 다운로드

## a. 파일 업로드

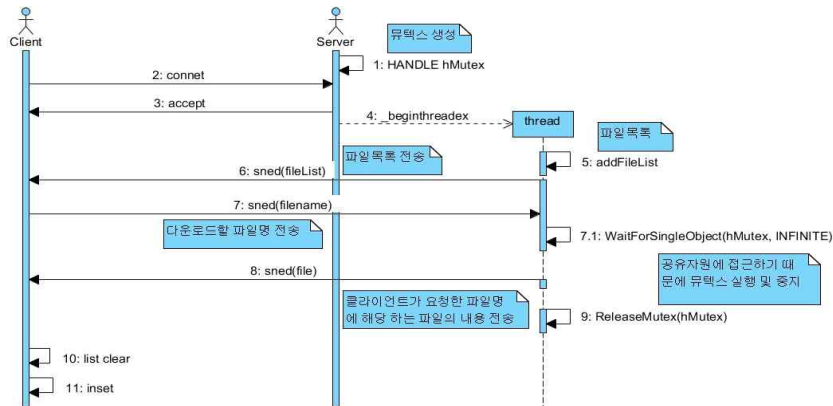


- 파일 업로드 시 클라이언트는 서버의 IP와 PORT정보를 포함하여 연결요청
- 연결 요청 수신 시 서버는 이를 승인하고 해당 클라이언트를 관리할 쓰레드를 생성
- 연결이 승인 되면 클라이언트는 파일의 이름과 저장 내용을 포함한 문자열을 서버로 전송
- 서버는 해당 문자열을 분리하여 저자, 글쓴이, 내용 순서로 file 구조체에 저장하며 file구조체는 연결리스트로 연결됨
- file(공유자원)에 접근 시 뮤텍스 락, 연락을 통해 상호배제 등 공유자원 문제를 해결

## - 실행화면



## b. 파일 다운로드



- 파일 다운로드 시 클라이언트는 서버의 IP와 PORT정보를 포함하여 연결요청
- 연결 요청 수신 시 서버는 이를 승인하고 해당 클라이언트를 관리할 쓰레드를 생성
- 서버는 파일의 자신의 파일 리스트에서 제목을 모두 더한 문자열(파일목록)을 클라이언트에게 전송
- 클라이언트는 자신이 다운받고자 하는 파일명을 서버로 전송
- 서버는 클라이언트가 요청한 파일명에 해당하는 파일을 자신의 파일 리스트에서 검색 한 후 해당 파일명과 일치하는 파일의 내용을 클라이언트로 전송
- 클라이언트는 파일을 수신 하면 자신의 리스트(이전에 입력한 문자)를 초기화 하고 수신한 문자열을 한 문자 씩 읽어 자신의 리스트에 저장
- file(공유자원)에 접근 시 뮤텍스 락, 언락을 통해 상호배제 등 공유자원 문제를 해결
- 실행하면

```

C:\Users\dmltj\OneDrive\Documents\2016년도 2학기\기초 C:\Users\dmltj\OneDrive\Documents\2016년도 2학기\기초
Input server IP : 192.168.219.106 Input port number : 20000
Input server port : 20000
Listening...
[File list] | [autor] <fileName>
Connected Client IP : 192.168.219.106
[kim] <data> Connected Client IP : 192.168.219.106
[choi] <c> Connected Client IP : 192.168.219.106
[park] <java> Connected Client IP : 192.168.219.106
Input you want file name : c
  
```

- 서버로부터 파일 목록 수신 및 다운받을 파일명 입력

```

C:\Users\dmltj\OneDrive\Documents\2016년도 2학기\기초프로젝트\프로젝트2\Client.exe
[F1]FileOpen [F2]FileSave [F3]FileUpload [F4]FileDownload [F5]Exit
asdskfkr
bkrkrknrk
krkmkr
  
```

- 전송한 파일명에 해당하는 파일의 내용 수신

## 2. 팀원 별 개발 역할 및 개발내용

팀원명	상세역할	주요 기여 내용
김의섭	문자 삽입	문자 입력 시 노드의 삽입, 삭제 기능 및 연결 상태 변화 구현
	문자 삭제	문자 삭제 시 노드의 삽입, 삭제 기능 및 연결 상태 변화 구현
	파일 업로드	파일 업로드 시 서버-클라 간 메커니즘 설계 구현
	파일 다운로드	파일 다운로드 시 서버-클라 간 메커니즘 설계 구현
	파일 다운로드 요청 처리	파일 다운로드 시 서버-클라 간 메커니즘 설계 구현
	공유자원 관리	서버에서 관리하는 파일(공유자원)을 동시에 여러 클라이언트(쓰레드)가 접근하지 못하도록 설계 및 구현
	포커스 추적	현재 포커스(노드의 입력, 삭제 위치)를 저장하기 위한 자료구조 구현
김경보	커서 출력	현재 포커스가 맞춰져 있는 노드의 위치로 커서 출력 및 방향키 end, home등의 기능키로 커서 위치 이동기능 구현
	파일 업로드	파일 업로드 시 서버-클라 간 메커니즘 설계 구현
	파일 업로드 요청 처리	파일 업로드 시 서버-클라 간 메커니즘 설계 구현
	파일 저장/불러오기	파일 저장 및 불러오기 기능 구현
	인터페이스	클라이언트에서 작업 전환 시 인터페이스 구현 및 설계

## 과제수행방법

## 1. 모듈화

- 각 기능을 크게 두 부분으로 분담하여 업무 진행
- 각 모듈의 부분의 문제들을 적게는 2가지 크게는 5가지의 부분으로 모듈화 하여 코딩
- 서버와 클라이언트 단위로 업무를 분담하지 않고 기능 단위로 업무를 분담  
=> 서버-클라이언트 구조 특성상 디버깅이 용이하게 하기 위해

## 2. 프로토타입

- 실행 가능한 프로토타입을 빠르게 작성하여 서로 의견을 교환
- 실제 프로토타입 -> 평가 -> 개선을 반복

## 3. 미팅

- 수업시간을(월3, 수12) 제외한 팀원미팅을 주 3회 이상 진행(주말)
- 수업시간을(월3, 수12) 제외한 교수님과 미팅을 주 2회 진행(주말)
- 미팅 시 서로 작성한 프로토타입 및 로직을 선보이고 의견 교환

\* 참여개발자간의 역할분담 및 추진 일정

주요내용	참여자	추진 일정 (주)				
		1	2	3	4	5
프로젝트 계획, 분석, 정의	김의섭 김경보	✓				
파일 업로드	김경보		✓	✓		
파일 다운로드	김의섭		✓	✓		
공유자원 관리	김의섭				✓	
인터페이스	김경보				✓	
시스템 통합	김의섭 김경보					✓

### 최종결과물

\* 기능 : 문서 공유 프로그램

\* 규격 : 입력 버퍼의 한계가 없는 클라이언트(연결리스트는 동적인 자료구조이기 때문에)  
파일을 저장 할 수 있는 버퍼의 한계가 없는 서버  
(각각의 파일은 연결리스트로 연결되기 때문에 버퍼의 한계가 없음)  
서버와 클라이언트 간의 메시지 송수신은 비동기 적으로 진행  
(하나의 클라이언트가 접속 해 있는 경우 다른 클라이언트의 접속도 허용하기 위해)

\* 특징 : 클라이언트(메모장 프로그램)는 4중 연결리스트를 활용하여 각 문자를 노드에 저장하고 리스트에 연결  
서버-클라이언트간의 파일 송수신 시 데이터는 문자열로 송수신이 일어남  
서버는 각각의 클라이언트에서 작성한 파일(텍스트파일)을 수신 하여 저장하고 수신한 파일들을 모아 파일리스트를 구성  
클라이언트는 서버에 있는 파일리스트 중 한 번의 요청에 하나의 파일을 수신할 수 있으며 서버의 파일리스트에 자신의 파일을 저장 가능  
서버가 관리하는 파일리스트는 동시에 여러 쓰레드(클라이언트)에서 접근 불가

\* 구조 : 클라이언트의 파일은 맨 처음 문자 앞의 head 포인터, 문서의 맨 마지막 엔 tail 포인터로 구성 되며 각 줄의 맨 마지막은 개행문자가 삽입되어 개행문자를 통해 문자열의 줄번호 구별이 가능  
서버는 클라이언트로부터 수신한 파일의 문자열 및 파일이름을 저장 할 수 있는 구조체를 가지고 있으며 해당 노드는 연결리스트로 연결되어 파일리스트 형성

### 결과활용계획

\* 본 프로젝트의 문서공유 프로그램을 통해 메모장으로 작성된 텍스트 문서를 사용자 간에 공유가 가능하게 함.

\* 문서공유가 가능하기 때문에 이 점을 살려 해당 문서에 대한 공동 작업이 가능하도록 그룹을 설정 할 수 있는 기능을 추가할 계획

\* 문서공유가 가능하기 때문에 다른 사용자가 작성한 텍스트파일을 다운로드가 가능 함 이 점을 살려 문서에 대한 평점 기능을 추가 하여 유용한 파일을 다운받을 수 있도록 할 계획

### 에로사항 및 건의사항

1. 에로사항( 프로젝트 개발 시 발생된 문제점)

1.1. 공유 자원 문제

- 서버에서 관리하는 파일의 목록의 경우 여러 쓰레드에서 접근하는 공유자원인데 이 공유자원을 보호해 주지 못해 문제가 생김, 해당 문제는 뮤텍스 락, 인락을 통해 여러 쓰레드가 동시에 공유자원에 접근하지 못하게 막음

2. 건의 사항 ( 향후 프로젝트 개발 시 필요한 지원 및 개선 사항에 대한 건의)

2.1. 버전 관리 부분에서 에로사항이 있었습니다. 이미 GitHub등의 환경을 대략적이라도 소개해 주셨으면 좋겠습니다. 기존 에 이러한 내용을 담은 강의가 개설되어 있지만 수강하지 못했거나 수강하지 않은 학생을 위해 배려해주셨으면 좋겠습니다.

2.2. 프로젝트의 규모가 기존에 진행하던 프로젝트에 비해 대규모인 경우가 많은데 이런 경우 약간의 개발 방법론 혹은 방법에 대해 대략적인 소개가 있었으면 좋겠습니다.