시스템프로그래밍
Proxy #3-1

담당교수 : 최상호 교수님(목4)

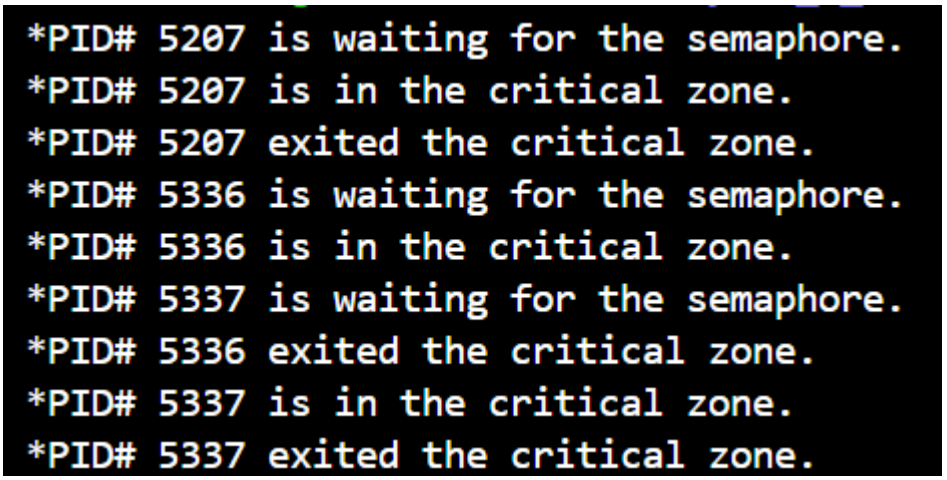
2021202003 강준우

Proxy #2-4

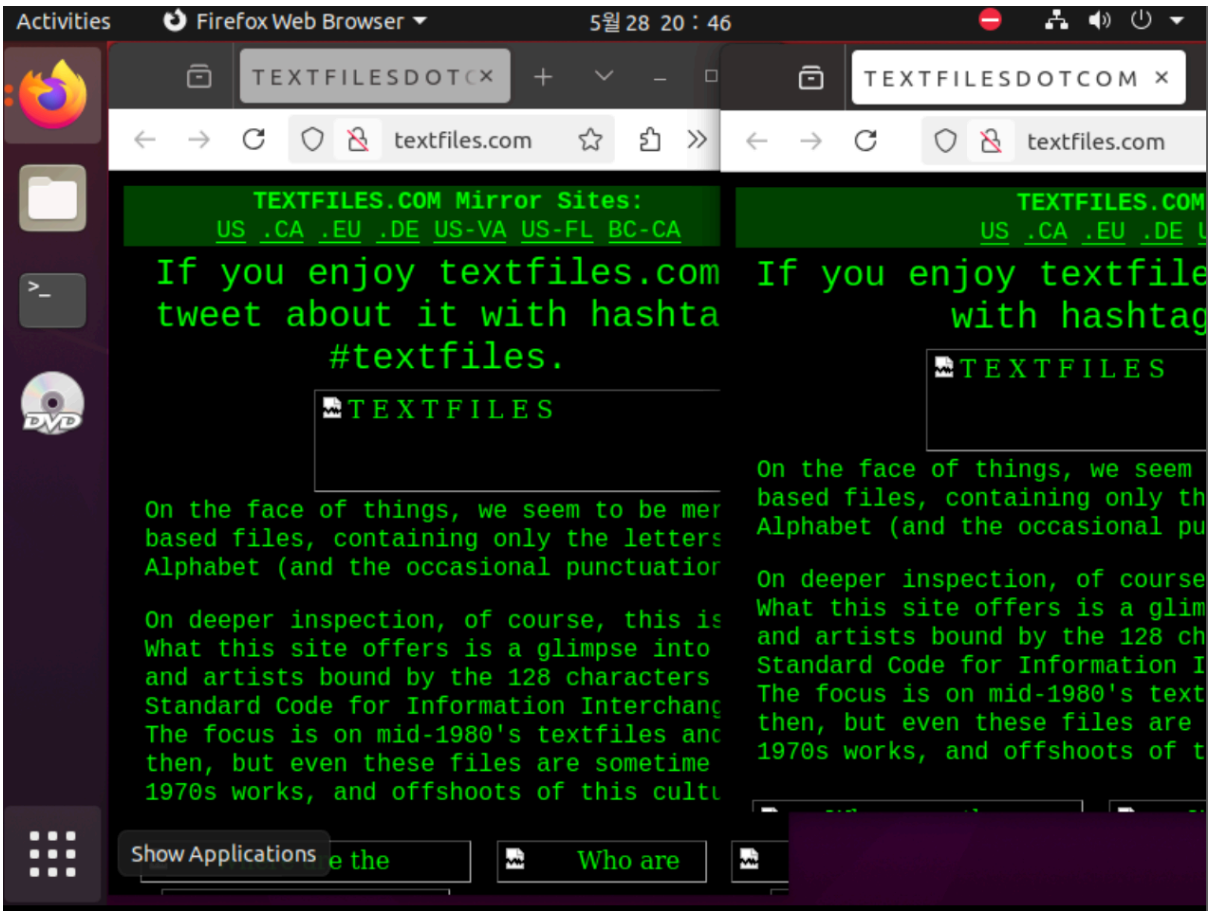
Introduction

이번 과제에서는 기존의 2-4과제에서 추가로 동시접근에 따른 제어를 추가하게 되었습니다. semkey값은 포트숫자와 같게 하며 추가로 한번의 프로세스에 따라 한번의 로그파일을 기록하게 설정하도록 요구됩니다. 하지만 접근시 터미널에서는 대기와 크리티컬 엑싯에 대한 내용들이 기록되어야 합니다.

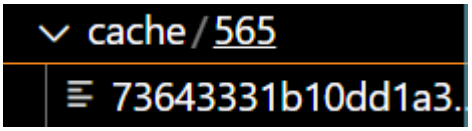
결과화면

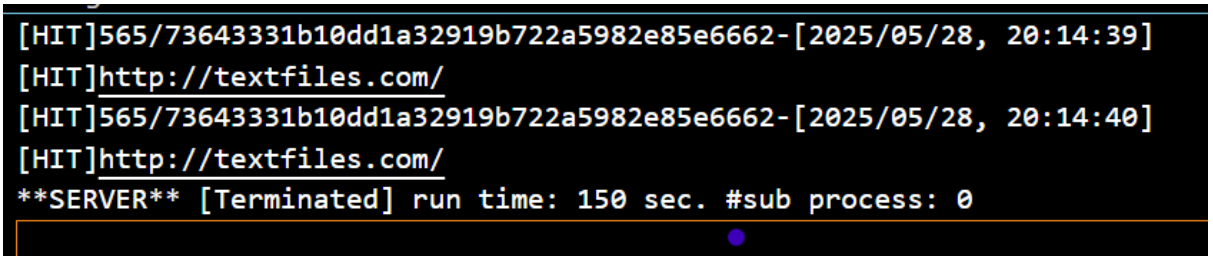


파이어폭스 페이지 2개를 연달아서 접속하여 시도해보았습니다.
이에따라 5336번과 5337번의 프로세스가 싫되지만 늦게 들어온 5337번은 대기하다가 이후에 처리되는 모습을 확인할 수 있었습니다.



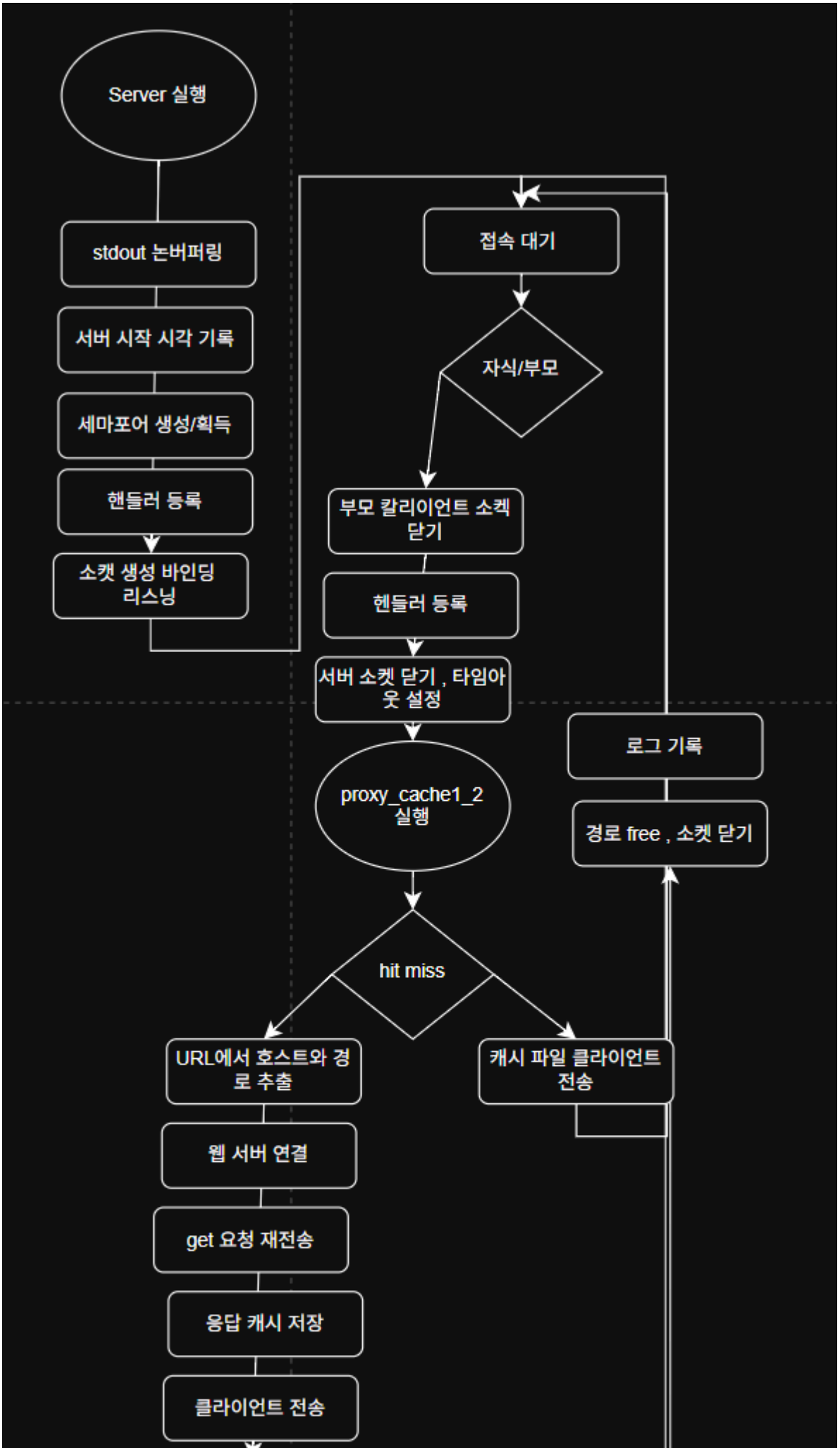
해당 결과 이전에 경로가 5207로 인해 생성되어 hit hit가 발생한 모습입니다.





마지막 두개의 로그파일을 확인한 모습입니다.

Algorithm – Flow Chart


Server 실행
stdout 논버퍼링
서버 시작 시각 기록
세마포어 생성/획득
핸들러 등록
소캣 생성 바인딩 리스닝
접속 대기
자식/부모
부모 칼리이언트 소켁 닫기
헨들러 등록
서버 소켓 닫기 , 타임아웃 설정
proxy_cache1_2 실행
hit miss
URL에서 호스트와 경로 추출
웹 서버 연결
get 요청 재전송
응답 캐시 저장
클라이언트 전송
캐시 파일 클라이언트 전송
로그 기록
경로 free , 소켓 닫기

psudo code

프로그램 시작
    표준 출력 버퍼링 해제
    서버 시작 시간 기록
    세마포어(동기화용) 생성 또는 이미 있으면 획득 (PORTNO 사용)
    시그널 핸들러 등록 (SIGINT: 종료 처리, SIGCHLD: 좀비 프로세스 방지)
    서버 소켓 생성, 주소 바인딩, 리스닝 시작

클라이언트 접속 대기 및 처리 (메인 루프)
    [반복] 클라이언트 연결 대기 (accept)
    연결 성공 시 자식 프로세스 생성 (fork)

자식 프로세스 분기
부모 프로세스:
    클라이언트 소켓 닫고, 다시 루프(새 연결 대기)
자식 프로세스:
    SIGALRM 핸들러 등록 (읽기 타임아웃)
    서버 소켓 닫음
    sub_process_count++
    15초 타임아웃 알람 설정

클라이언트 요청 처리
    클라이언트로부터 요청 메시지 읽기
    요청 메시지에서 HTTP 메소드, URL, 프로토콜 파싱
    메인 리소스인지 확인

캐시/로그 동기화 처리 - proxy_cache1_2
    캐시 폴더, 하위폴더, 파일 경로 생성(없으면 생성)
    파일이 이미 있는지(캐시 HIT 여부) 확인
    세마포어 P 연산: 진입 대기

    크리티컬 섹션
        1초간 sleep
        로그 파일에 HIT/MISS 기록
        세마포어 V 연산: 진입 종료
    캐시 파일 경로 리턴

HIT/MISS 분기 응답
캐시 HIT 시:
    캐시 파일을 열어 클라이언트로 전송

캐시 MISS 시:
    URL에서 host, path 추출
    웹서버 DNS 조회, 소켓 연결
    HTTP GET 재전송

웹서버 응답을 캐시에도 저장, 클라이언트에도 전송

리소스 정리 및 종료
동적 메모리 해제, 클라이언트 소켓 닫기
자식 프로세스 종료 (exit)

서버 종료 처리(SIGINT)
서버 종료 시 서버 실행시간, 프로세스 수 로그 기록
세마포어 삭제(IPC_RMID)

고찰

이번 과제에서는 세마포어를 통해서 프로세스의 동시접근을 관리 할 수 있었습니다.
특히 동시접근에 있어서는 대기 및 수행등을 처리해야하는 부분이 있었습니다.
함수중 프록시 1-2라는 함수가 있지만 사실상 계속 변경하다보니 더이상 1-2가 아닌 느낌을 받기도 했습니다. 함수 이름을 만들 때 규칙등을 좀더 연습해봐야 한다는 생각을 하게 되었습니다.