김영찬

kinkibx@gmail.com
010-8769-9038

근본을 중요시하고 납득될 때까지 물음표를 던지는 신입 개발자 김영찬입니다.

보다 나은 성능의 개선과 새로운 기술에는 근본적인 지식과 이해가 선행되어야 하고, 이는 수많은 기술들과 빠르게 변해가는 세상에서 무엇보다 중요한 것이라고 생각합니다.

동작 원리를 추측하고 기초지식과 이를 바탕으로 배우고 문제를 해결하려고 노력하는 개발자를 목표로 하고 있습니다.

학력

인하대학교 학사
사회인프라공학 전공
2016년 3월 - 2023년 2월

어학

OPIC - Intermediate High
(2022.10)

교육

42Seoul 라피신 및 본과정
2023년 2월 ~ 진행중

기술

Language	●C ●C++ ●Python ●Assembly
Library / Framework	●Django
Tools	●Nginx ●Docker ●ELK(ElasticSearch/Logstash/Kibana)
Etc	●Linux ●LLDB

●원활한 활용
●활용가능
●사용가능

프로젝트

프로젝트명	Webserv
과제목표	정적파일 서빙과 CGI처리가 가능한 웹서버 구현(Request파싱 및 CGI처리 담당)
내용	- TCP/IP Socket관련 함수를 이용해 HTTP프로토콜 웹서버 구현 - RFC문서를 통한 HTTP 프로토콜의 기본적인 이해 - Multiplexing I/O 모델 kqueue을 사용하여 한번에 한 이벤트(read, write), 한번에 일정 버퍼사이즈만큼만 읽게 해서 요청들의 공평한 처리 - 리퀘스트 파싱 시에 헤더들의 유효성검사 및 길이검사로 유효성판단 - apache처럼 CGI방식으로 사용하여 FAST-CGI 방식보다 서버부담이 커짐 - TIMER 이벤트 플래그를 활용하여 비활동적인 클라이언트에 따른 메모리 낭비 감소 및 CGI의 비정상적으로 긴 처리를 없앰 - 클라이언트 하나당 fd 하나에 이론상 여러 cgi 프로세스가 생길 수 있고, 하나의 CGI 처리에 필요한 fd가 파이프2개와 PID 등의 정보가 필요하므로 이를 연결시켜주면서 동시에 리퀘스트 리스트를 순회하며 찾는 시간을 줄이기위해 Map컨테이너와 배열을 이용해 모든 자료 검색을 O(logn)에 가깝게 처리하여 테스터기 소요시간 줄임 - CGI와 리스폰스 작성시, 파이프 버퍼사이즈와 소켓버퍼사이즈의 한계 상 여러번 쓰이고 읽어야 하는데, 이 소요시간을 줄이기위해 레퍼런스 사용 및 따로 읽고 쓴 정수형 포인터를 저장해놓음으로써 시간을 5배 이상 단축시킴
세부내용 (Github)	https://github.com/dudcks0994/Webserv

프로젝트명	Transcendence
과제목표	Django와 VanilaJS를 이용한 핑퐁게임 웹사이트 구현 (Backend 및 Docker파트 담당)
내용	- 웹소켓 프로토콜에 대해 기본적인 이해, 비동기 프로그래밍 - 웹소켓을 이용하여 친구의 온라인/오프라인 스테이터스를 구현 - JWT토큰을 이용한 로그인을 구현한 상태에서 OTP인증 추가시에, OTP코드를 입력할 동안 회원 식별을 위해 Session인증만을 활용함 - Docker을 사용하여 Nginx를 프록시버로 백엔드와 프론트서버 연결 및 self-signged TLS를 적용하여 HTTPS 통신적용 - 세 개의 도커 컨테이너를 이용해 ELK스택을 구축하여 로그 시각화 및 인덱스의 생명주기 설정
세부내용 (Github)	https://github.com/dudcks0994/transcendence

프로젝트명	Philosopher
과제목표	멀티쓰레드/멀티프로세스로 기아회피 및 데드락방지하는 '식사하는 철학자' 문제 구현
내용	- 뮤텍스 / 세마포어를 통해 포크를 집는 행위의 원자성 보장 및 데이터레이스 방지 - 데드락 방지를 위해 짝수번째 철학자들이 먼저 먹을때 잠시 홀수번째 철학자를 기다리게하고 같은 방향 포크를 먼저 들게 하여 번갈아가면서 먹을수 있게 처리 - 멀티프로세스 환경에서 먹는횟수를 공유하기 위해, 각각의 감시용 스레드를 생성한 뒤 먹는 루틴 사이클에 철학자 한명씩 세마포어를 만들어 post, wait하면서 카운트하게 처리 - 철학자가 죽은 이후 출력이 나오지 않게 출력 뮤텍스를 만들어 죽음체크-출력의 원자성 보장
세부내용 (Github)	https://github.com/dudcks0994/Philosoper

프로젝트명	Minishell
과제목표	기본적인 기능들을 구현한 미니 Shell 프로그램 만들기
내용	- POSIX 문서를 참고해 입력을 파싱하여 메타캐릭터를 기준으로 토큰화 하여 연결리스트로 만들고 토큰의 타입을 저장하여 각각 처리 - 자식프로세스를 생성해야하는 heredoc과 명령어 실행시, 시그널입력이 부모프로세스에게까지 전달되는것을 막기위해 SIG_IGN을 활용 - 부모프로세스의 모든것을 복사하는 fork()의 특성 상, 파이프 생성에 따른 프로세스 최대fd 제한을 피하기위해 파이프 두개를 이용해 복사 및 닫기로 만개 이상의 cmd도 실행가능하게 처리 - 입력을 기다리며 종료되지 않는 프로그램의 경우 SIGPIPE가 발생하게 적절하게 닫아주어 정상적인 입출력이 가능하게 처리 - 입력에 따라 토큰갯수와 명령어갯수, 파이프갯수에 따른 파라미터에 필요한 메모리가 달라져, malloc으로 동적메모리 할당 및 사용종료 후 free로 메모리해제
세부내용 (Github)	https://github.com/dudcks0994/minishell