

cmake 설정

- 관리기능
- 서브 디렉토리

동적할당

- 빠질 수 없는 기능
- 설정하면 빼는것도 항상 하기

기타문법

- 공용체 union
- 열거형 enum

booksql

- 기능 4개

=====

분할컴파일

표준입출력

표준파일 입출력(fd)파일 디스크립터(File Descriptor)

- 0. stdin
- 1. stdout
- 2. stderr

	[버퍼]	열거형
stdin	O	0
stdout	O	1
stderr	X	2

read() 형식으로 읽어 올 때(scnaf, get)

- 버퍼가 데이터가 없으면, 대기(블록킹 함수)
- 버퍼에 데이터가 있으면, 그만큼 읽어서 return

scanf()가 대표적인 예.

fscanf()는 fd를 명시해서 쓸 수 있음

get()시리즈: 버퍼를 비우지않음

getc() - fgetc(): char 하나씩 리턴

gets() - fgets(): enter 입력될 때까지 모은 것을 리턴

f가 붙은 get 애들은 fd에 대한 명시가 필요함->함수로 정의

하지만 getchar()를 비롯한 나머지는 fd에 대한 명시가 없어도 됨->매크로로 정의

입력 버퍼와 출력 버퍼는 따로 운영
stdout 출력버퍼는 정상작동해도
stdin 입력버퍼는 f플러시?에대해 복잡하다

출력버퍼에 쌓이면 화면으로 출력
-> 내부 로직에 따른 출력
-> \n 개행문자로 인한 앞부분 출력

stdout
버퍼에 write()를 통해 넣을 수 있음

대표적인 예시로 printf()
fprintf()는 fd를 명시해서 쓸 수 있음
put()
putc() -> 매크로로정의
fputc() -> 함수로정의 -> putchar() 파생됨

putchar의 사용처
- 파일(하드)에 직접 저장할 수 있음
- 네트워크 스트림 fd

출력의 블록킹이 나오는 원인은 1가지: 버퍼가 꽉 찼을 때

버퍼에 쌓아두고 능력대로 처리하니 처리가 조금 빠름
IoT 통신(시리얼, tcp)에 통신 버퍼가 있음

put이나 printf로 통신버퍼에 저장함
IoT기기에서 wifi모듈의 통신 버퍼에서 전송
들어온 통신버퍼를 임시로 저장함
출력 버퍼로 옮긴 후 모니터로 이동

월요일 16시에 c언어 시험
//볼링 프로그램 만들기
//balling.c main.c balling.h
볼링 스코어판 제작

1. 점수를 입력함수 프레임마다 10프레임까지
각 프레임줄마다 300점까지 계산

2. 1번 테이블[3] 사용자명, 점수, 날짜----->score table
2번 테이블[2] 사용자명, 월별 횟수, 날짜(월)-----> counting table
==>mysql
토탈 테이블 사용자명, 점수, 날짜, 게임횟수 카운트
아마 join으로 테이블 엮을듯

3. 1인용