

전자공학도를 위한 프로그래밍 2019 프로젝트 II

공동주택 택배 보관함 구현

1. 문제 정의

- 아파트와 같은 공동주택 단지로 전달되는 택배를 보관하는 보관함 프로그램 구현
- 공동주택 단지 및 택배와 관련해서 아래와 같이 가정을 함

- 단지는 1동부터 N_BUILDING 동의 건물로 구성되며, 모든 건물은 모두 동일하게 1층부터 N_FLOOR층까지, 각 층은 N_ROOM개의 집이 존재
 - n층 m번째 집의 호수는 $n*100+m$ 으로 표기 (예시 : 3층 8번째 집 -> 308호)
- 택배는 공동주택 단지 내 한 집에 전달되는 것으로, 그 내용은 문자열 형태의 메시지임
 - 받는사람 주소는 건물 동수와 호수로 표기
 - 메시지 내에는 공백글자 포함 불가

- 위의 가정 하에서, 구현하고자 하는 택배 보관함 시스템은 아래 기능을 지원해야 함

- 택배 보관함은 N 개의 row와 M 개의 column으로 구성된 storage를 가짐
 - N, M 값은 설정 파일로부터 읽어와서 정의
- push package : 비어있는 storage 중 하나를 사용자가 선택해서 택배를 넣음
 - 입력 전에 택배 보관함 상태를 사용자에게 표기
 - 사용자로부터 받는사람 주소, 비밀번호를 입력받아 저장 및 설정
 - 사용자로부터 택배 내용 문자열을 입력받음 (단, 입력에 공백글자가 섞여 있으면 공백글자 직전까지만 저장)
- extract package : 사용자가 택배를 꺼내서 내용 확인
 - 꺼내고자 하는 storage 좌표를 입력하고, 비밀번호 입력
 - 입력된 비밀번호를 대조 후 택배를 꺼내면서 택배 내용 문자열을 화면에 표기
- check status : 사용자가 보관함의 storage들의 상태를 확인
 - 모든 storage에 대해 택배가 있는지 여부와 택배 존재 시 받는사람 주소를 출력
- find my package : 사용자가 자신의 택배가 어디있는지 확인
 - 내 주소(동호수)를 입력하면 택배가 있는 모든 storage들의 좌표를 화면에 출력
- Master 비밀번호 기능 지원
 - 오래된 택배를 처리할 때, 어떤 storage에도 master 비밀번호를 입력하면 무조건 택배를 꺼낼 수 있음
 - Master 비밀번호는 설정 파일로부터 읽어와서 정의

- 택배 보관함 시스템 프로그램은 아래와 같이 설정 파일을 활용해야 함

- 택배 보관함 상태 및 설정은 프로그램 실행 초반에 설정파일로부터 가져오고, 수시로 보관함 상태를 파일에 저장
- 설정 파일의 format은 아래와 같음 (총 K개의 택배가 보관되어 있는 상태를 가정)

N M

(Master 비밀번호 4자리)

(첫번째 택배 항목)

(두번째 택배 항목)

...

(K번째 택배 항목)

- 여기서 (k번째 택배 항목)이란 이전에 저장된 택배 보관함 상태를 기술한 것으로, 아래 형태임

(storage의 row번호) (storage의 column번호) (받는사람 동) (받는사람 호) (비밀번호) (택배내용 문자열)

- 보관 상태가 변경될 때마다 설정 파일을 새로 write함
- 프로그램 시작 시 설정파일에 저장된 이전 보관 상태를 택배 보관함 DB에 복원

- storage.c 구현 시 아래를 반드시 지켜야 함

- storage_t 구조체 선언은 수정 없이 그대로 사용해야 함
- main 함수가 정상 동작하도록 storage.h 내에 prototyping된 함수를 모두 구현해야 함
 - parameter, return 자료형 등은 변경 가능하나, 그경우 그에 맞게 main.c도 수정해야 함
- 택배 보관함과 관련된 내용 (보관된 택배 관련 정보나 택배보관함 크기 등)은 storage.c 내 static 변수 형태로 저장이 되도록 구현되어야 함

2. 코딩 방향

- 주로 storage.c의 비어있는 부분을 구현하면 되며, 되도록 main.c 코드는 그대로 두는 것을 추천

- storage.c 내의 아래 함수에 대해 구현

- initStorage() : 특정 보관함 하나를 초기화 함
- inputPasswd() : 특정 보관함에 대해 비밀번호를 입력 받아서 맞는지 확인
- str_backupSystem() : 현재 보관함들의 상태 및 설정 값들을 파일에 저장
- str_createSystem() : 택배보관함 구조체 자료구조 생성
- str_freeSystem() : 택배보관함 자료구조 메모리 해제
- str_pushToStorage() : 특정 보관함에 택배 넣기
- str_extractStorage() : 특정 보관함에서 내 택배 꺼냄
- str_findStorage() : 내 택배가 있는 보관함 찾기

- 함수나 변수 정의, 코드 흐름에 대해 다른 사람들이 알아볼 수 있도록 주석을 삽입

- 기능 혹은 일정 코드부분을 구현하거나 디버깅으로 코드수정 때마다 형상관리에 올려서 변화 추적할 수 있도록 함

3. 배점 및 기준

- 프로그램 동작 점수 : 100점

- 1번 문제 정의에 명시된 기능들이 충실하게 잘 동작하는지 여부
- 부분적으로 동작하면 그에 맞게 부분점수 부여 예정

- 코드 관리 점수 : 50점

- 주석을 충실하게 달았는지 여부 (함수 및 변수 정의에 대한 설명 및 코드 흐름에 대한 설명)
- 들여쓰기를 잘했는지 여부
- 메모리 관리가 잘되는지 여부 (memory leakage, memory corruption 여부 확인 예정)
- 형상관리에 코드를 구현한 이력이 잘 남아있는지 여부

- copy 여부에 대해 집중적으로 볼 예정이며, 적발 시 관련자 모두 0점으로 처리

4. 결과물 제출 방법

- 제출은 각자 사용하는 github ID로 형상관리 서버에 올림

- deliverySystem이라는 이름의 새로운 repository를 생성해서 올림
- 기한 시간 직전에 올라간 코드를 기준으로 채점
- 형상관리 서버에 deliverySystem이라는 repository 코드가 없으면 제출하지 않은 것으로 간주할 예정이므로, 제출 후 web site상에서 본인이 올린 코드가 제대로 보이는지 반드시 확인 요망

- 제출 기한 : 12월 12일 목요일 자정까지