**2021년 2학기**

**프로그래밍과 문제해결**

**Assignment #4**

**담당교수: 윤은영**

**학번: 20217016**

**학과: 무은재학부**

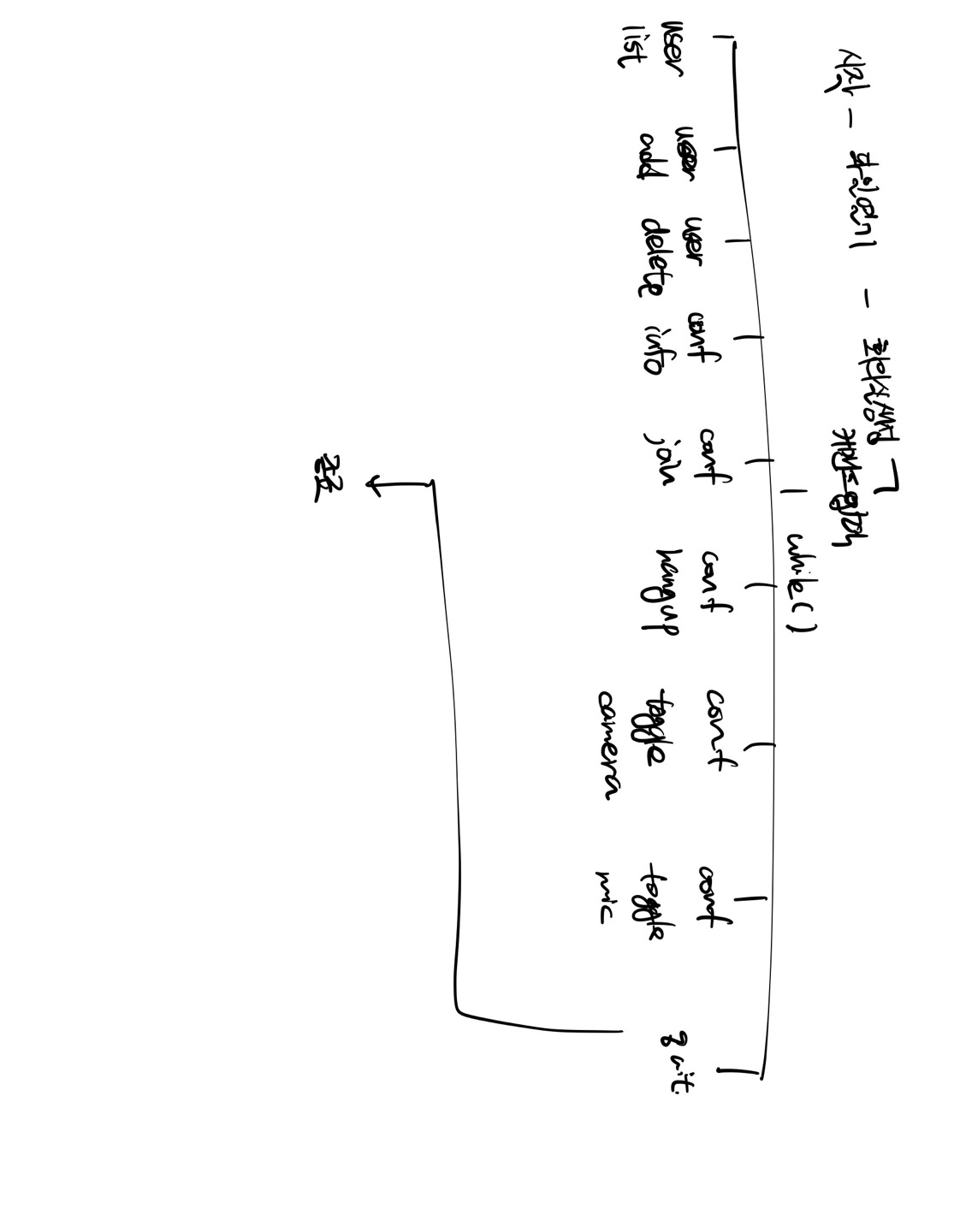
**이름: 최대현**

**POVIS ID: daehyeonchoi**

**명예서약(Honor Code)**

**“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”**

**Sturcture Chart**

이번 화상 회의 관리 프로그램의 structure chart는 다음과 같다 

**알고리즘**

* **Psuedocode**

1. 파일 명을 입력받고, 읽기 모드로 열어 데이터를 저장한다. .
2. Command를 입력받는다.
3. Command를 맞게 입력했다면 그 커맨드에 맞는 기능을 수행한다.

3-1. ‘user list’를 입력하면 현재 user들의 정보가 포함된 리스트를 출력한다.

3-2. ‘user add’를 입력하면 userlist에 새로운 user를 삽입한다.

3-3. ‘user delete’를 입력하면 userlist에서 특정 user를 삭제한다.

3-4. ‘conf info’를 입력하면 현재 회의실의 정보를 출력한다.

3-5. ‘conf join’를 입력하면 회의실에 특정 user를 추가한다.

3-6. ‘conf hangup’을 입력하면 회의실에 있는 특정 participant를 회의실에서 제거한다.

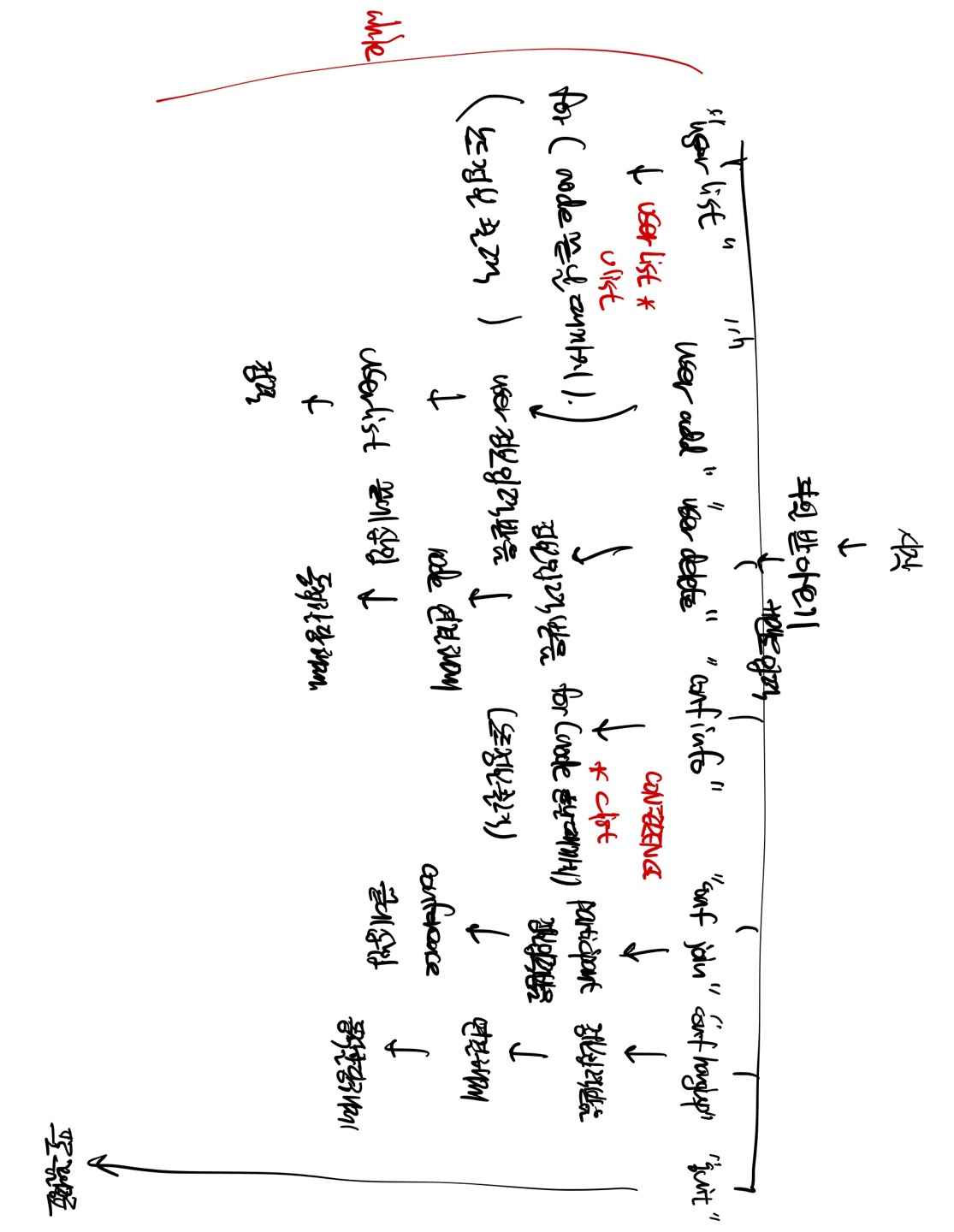
3-7. ‘conf toggle mic’를 입력하면 회의실에 있는 특정 participant의 마이크를 켜거나 끈다.

3-8. ‘conf toggle camera’를 입력하면 회의실에 있는 특정 participant의 카메라를 켜거나 끈다.

1. 기능을 수행한 다음 command를 다시 입력받고, 이 과정을 계속하여 반복한다.
2. ‘Quit’ command를 입력할 경우 프로그램이 종료된다.

* **Flowchart**

이번 화상 회의 관리 프로그램의 대략적인 Flowchart는 다음과 같다.



**Data Structure**

텍스트이(가) 표시된 사진

자동 생성된 설명

기존에 주어진 user, participant, conference 이외에 구현의 편의를 위해서 userlist라는 구조체를 하나 더 만들어서 사용하였다. 그 안에는 conference의 형식과 유사하게 user의 수를 저장하는 변수와 head를 가리키는 user 포인터형 변수가 선언되어 있다.

**함수 설명**

1. **void print\_user\_list(USERLIST\* ulist)**

* userlist를 출력하는 함수이다. for문으로 출력하고, user 포인터 temp 변수를 선언하여 temp -> next 값이 NULL(즉, linked list의 끝을 의미)이 아닌 동안 userlist를 출력한다.

1. **void user\_add(USERLIST\* ulist)**

* userlist에 user을 추가하고, 사전식으로 정렬하는 함수이다. 처음부터 userlist의 올바른 위치에 넣는 알고리즘도 생각해봤지만, 끝에다 일괄적으로 노드를 삽입한 후 함수를 종료하기 전에 dic\_sort 함수를 통해 정렬하는 것이 더 편할 것이라고 판단했다.

1. **void user\_delete(USERLIST\* ulist)**

* userlist에서 user를 제거하는 함수이다. Strcmp 함수를 통해 원하는 위치를 찾고, 그 위치에 해당하는 노드를 linked list에서 제거한 뒤 나머지 node를 이어준다. User delete 함수를 통해 제거된 노드는 바로 동적 할당을 해제해주어야 한다.

1. **void conf\_info(CONFERENCE\* clist)**

- 현재 회의실의 이름, 인원, 명단 등을 출력하는 함수이다.

1. **void conf\_join(CONFERENCE\* clist, USERLIST\* ulist)**

* 회의실에 userlist에 포함된 user 중 한 명을 참가시키는 함수이다. Userlist에 있는 user인지 확인하기 위해 USERLIST\* 형 매개변수 ulist를 받아서 사용한다. CONFERENCE->participants->next에 해당하는 head 노드부터 한 칸씩 뒤로 가며 linked list의 끝을 찾아준 뒤 그 끝에 participant를 넣어준다.

1. **void conf\_hangup(CONFERENCE\* clist)**

* 회의실에서 participant를 제거하는 함수이다. 제거할 participant의 이름을 입력 받아 for문을 돌리면서 회의실에 있는지 검사하고, 있다면 그 노드를 제거한 뒤 나머지 노드를 잇는다.

1. **void toggle\_camera(CONFERENCE\* clist)**

* participant 중 한 명을 지정하여 카메라를 켜거나 끄는 함수이다.

1. **void toggle\_mic(CONFERENCE\* clist)**

* participant 중 한 명을 지정하여 마이크를 켜거나 끄는 함수이다.

1. **void dic\_sort(USERLIST\* ulist)**

* userlist 속 user들의 이름을 사전식으로 정렬하는 함수이다. , username과 usermail을 임시로 저장할 문자열 변수를 선언하여 버블 정렬과 유사하게 함수를 구현하였다.

1. **void insert\_user(USERLIST\* ulist, USER\* uPre, char \*sen)**

* user node를 userlist의 맨 끝에 붙이는 함수이다. 처음에 파일을 불러올 때와 나중에 user add 함수에서 동일하게 사용하기 위해 (이름 \t 메일) 형식의 문자열을 char 포인터 형으로 가져온다. 그 뒤 문장을 쪼개서 이름과 메일을 user -> name, user ->email에 strcpy 함수를 통해 저장한다. 그 뒤 ulist로 시작되는 linked list의 끝을 찾은 뒤 노드에 삽입한다.

1. **del\_user(\*USERLIST ulist)**

* userlist에 있는 모든 user node에 대해 동적할당 해제하는 함수이다.

1. **del\_conf(\*CONFERENCE clist)**

* conference에 있는 모든 participant node에 대해 동적할당 해제하는 함수이다.

**프로그램 실행 방법 및 예제**

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램을 실행하면, 유저 리스트 파일 이름을 입력해주세요 라는 문구가 뜨며 파일 명을 입력받는다. 문제에서 주어진 예시 파일인 user.txt를 입력하면, 파일을 읽기 모드로 정상적으로 열 수 있으며 “이름 \t 메일” 형태로 주어진 데이터를 불러오고, 그것을 userlist로 저장하여 user 몇 명을 불러왔는지 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

회의실 이름과 최대 사용자 수를 입력 받으면 회의실이 만들어졌다는 문구를 출력한 뒤 커맨드를 입력받는다. 커맨드의 종류는 총 9가지이며 만약 커맨드에 해당하지 않는 문장을 입력할 경우 다시 입력 받는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**1) user list**  텍스트이(가) 표시된 사진

자동 생성된 설명

user list 커맨드를 입력받으면 현재 user list에 저장된 user들을 차례로 출력한다. 이 때 순서는 ‘이름을 사전식 정렬’한 순서를 따른다.

**2)user add**

user add 커맨드를 입력하면 추가할 유저의 이름과 이메일을 입력받고, 역시 알맞은 위치(사전식으로 정렬된 위치)에 user를 삽입한다. 텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

삽입한 뒤 user list 커맨드를 입력하여 userlist를 출력하면 다음과 같다.

**3)user delete**

userlist에 있는 user를 제거하는 함수이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

커맨드를 입력하면 제거할 user의 이름을 입력 받고 제거할 것인지 한 번 더 물어본다. N를 입력하면 다시 커맨드를 입력 받고, y를 입력하면 제거했다는 안내 문구와 함께 함수가 종료된다. 다시 user list를 출력해보면 다음과 같이 Jin이 제거된 것을 볼 수 있다. 이 때, 제거 함수를 종료하기 전에 동적 할당을 해제하는 절차가 필요하다.

**4)conf info**

회의실 정보를 출력하는 함수이다. 아무 것도 하지 않은 채 conf info를 출력하면 회의실이 비어있으므로 서식만 출력되고, conf join 커맨드를 통해 회의실에 participant를 추가하는 절차가 필요하다.



**5)conf join**

회의실에 특정 user를 추가해주는 함수이다. 먼저 추가할 user의 이름을 입력받고, user가 userlist에 있다면 함수가 정상적으로 호출되면서 user의 이메일과 이름을 받아온다. 그 뒤 카메라와 마이크를 켠 상태로 시작할 지 정보를 입력 받아 저장한 후 함수를 종료한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**6)conf hangup**

회의에 참여하고 있는 participant 중 한 명을 회의실에서 제거하는 커맨드이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**7)conf toggle camera**

회의에 참여하고 있는 participant 중 한 명의 마이크의 on/off 여부를 바꾸는 커맨드이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다음과 같이 conf join 커맨드를 통해 회의실에 Sunny라는 유저의 카메라와 마이크를 켠 상태로 추가하였고, conf info 커맨드를 통해 출력한 현재 회의실 상황은 다음과 같다.

만약 participant의 마이크가 켜져 있다면, 그 participant의 마이크를 끄고, 꺼져 있다면 켜는 기능을 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**8)conf toggle mic**

카메라와 똑같은 알고리즘을 가진 함수를 호출하는 커맨드로, 역시 마이크가 켜져 있다면 끄고, 꺼져 있다면 켜 주는 커맨드이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**9)quit**

프로그램을 종료하는 커맨드이다. 모든 노드에 대하여 동적할당 해제를 완료한 뒤 프로그램을 종료한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**토론**

어떤 부분에서 동적 할당을 해야 하고, 해제해야 하는지 생각하는 것에서 시간이 조금 걸렸던 것 같다. 링크드리스트에서 연결을 끊기 전에 동적할당을 해제하고, 마지막에 링크드리스트에 남아있는 채로 프로그램을 종료하지 않기 위해 많은 시간을 투자했다.

**결론**

이번 assignment를 통하여 linked list를 이해할 수 있었다. 특히 문제에서 주어진 데이터구조를 벗어나서 나만의 프로그램을 구현하기 위해 새로운 자료형을 정의하고 구조체를 새로 만들어서 사용한 부분이 인상적이었다.

**개선 방향**

사용자 정의 함수를 더 많이 사용해서 코드의 가독성을 높이고 코드를 더 간단하게 만들 수 있을 것 같아서 개선해보고 싶다.