2021년 2학기

**프로그래밍과 문제해결**

**Assignment #4**

담당교수: 윤은영

학번: 20210370

학과: 무은재학부

이름 권민경

POVID ID: mkkwon

명예서약(Honor Code)

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

**구조체와 연결리스트를 이용한 화상 회의 관리 프로그램**

1. **문제의 개요**

* 파일 포인터를 통해 사용자를 입력 받고 사용자를 회의에 추가한다.
* 회의 참가자를 삭제하거나 지울 수 있고, 회의 정보, 참가자의 카메라 상태, 마이크 상태를 바꿀 수 있다. 이때, 구조체와 연결 리스트를 이용해야 한다.

화상 회의 관리 프로그램 구조

입력부: 처음 메인 화면에서 무슨 파일을 읽어올지 입력 받는다. 그리고 이후 회의 이름입력, 회의 최대 인원 입력, 명령어 입력을 받고 각 명령에서 요구하는 기능을 실행하기 위해 명령어를 입력 받는다.

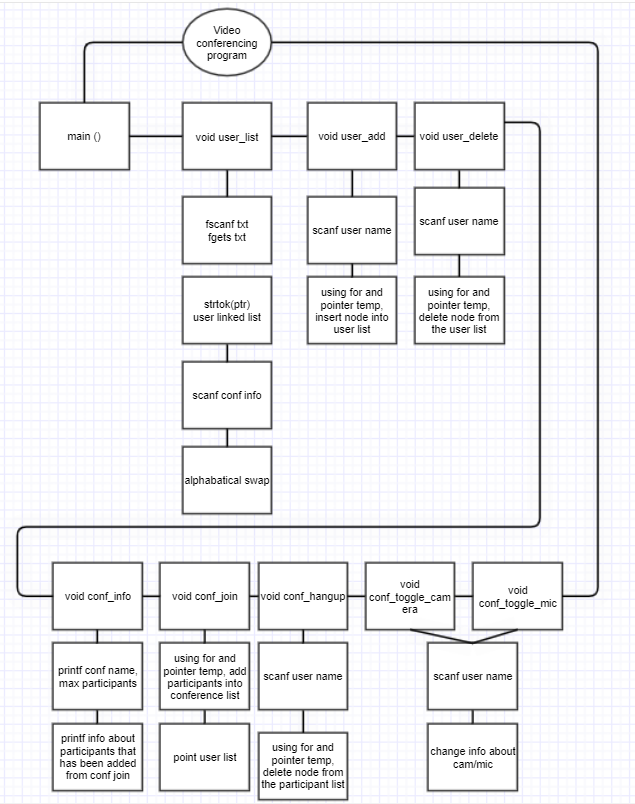
처리부: 처음 유저 파일을 읽어와 그 정보를 저장한다. 이때 데이터의 구조는 linked list로 구성하여 user list를 만든다. 이후 회의 관리 부분에서 participants를 linked list로 구성하여 participant list를 만든다. 또한 participants 삭제, 추가 카메라 켜기, 끄기 부분을 담당한다.

출력부: 유저 리스트, 참가자 리스트를 출력하며 회의 정보를 출력한다. 또한 각 명령을 수행할 때 요구되는 질문을 출력한다.

1. **알고리즘**

본 프로그램 작성을 위해 Pseudo코드 형태로 나타내면 다음과 같다.

|  |
| --- |
| Pseudo-algorithm for video conferencing program |
| //변수는 선언된 것으로 가정   1. Include essential headers <stdio.h>, <stlib.h>, <stdbool.h> and <string.h> 2. Print command screen, asking which file you want to open 3. Scan one sentence from txt and strtok name and email 4. Scan conference name and max participants of the conference 5. There are five programs required in this program 6. For user list, print all users 7. For user add, add one user, using linked list 8. For user delete, delete one user, using linked list 9. For conf info, print information about the conference and the participants 10. For conf join, add participants into the conference list 11. For conf hangup, delete participants from the conference list 12. for toggle mic and camera, turn off for whose cam/mic are on vice versa 13. allocation free for all variables that has be allocated |

위의 의사 알고리즘을 flow chart를 통해 표현하면 아래와 같다.

1. **프로그램 구조 및 설명**
2. 읽고 싶은 파일명을 스캔해온다

**프로그램 시작화면**

1. Fscanf를 통해 명령한 파일을 입력 받음.

파일의 정보를 한 문장 씩 스캔해오고 strtok를 사용해 이름과 이메일을 순서대로 자른후 participants list에 넣는다. 이때 participant list를 할당하고 연결하며, 포인터 변수를 사용하도록 한다.

1. Conference 정보를 입력 받는다.

회의 명은 무엇인지, 최대 참여 인원은 몇명인지 정보를 입력 받는다. 이때, 정보들을 conference list에 저장하도록 한다.

**유저 관리**

1. User list

파일에서 스캔해온 모든 유저 정보를 프린트한다. 이때 temp를 사용하여 노드를 하나씩 옮기면서 정보를 출력하도록 한다. 노드를 하나씩 옮길 때 반복문을 사용하도록 한다.

1. User add

추가하고 싶은 유저의 이름과 이메일을 입력 받아 node에 넣는다. 이때 만약 중복되는 유저가 있다면, 경고 메시지를 출력한다.

1. User delete

삭제하고 싶은 유저의 이름을 입력 받아 삭제한다. 이때의 알고리즘은 랩 과제를 참고하여 temp로 삭제하고 싶은 유저의 node를 찾고 포인터를 지우고 연결하여 구상하도록 한다.

**회의실 관리**

1. conf info

conf join에서 추가한 participant리스트와 회의 정보를 출력한다. 마찬가지로 반복문과 포인터를 사용한다.

1. conf join

user list중 회의에 추가하고 싶은 사람의 정보를 입력 받고 node를 만들어 마이크 카메라 on off 여부를 판단한다. 이후 user list에서 사람의 정보를 찾아 포인터로 가리킨다.

1. conf hangup

participant list에서 입력한 유저를 삭제한다. 이때 알고리즘음 user delete와 유사하다.

1. conf toggle camera/mic

conference 리스트에 저장된 참가자 정보에 camera와 mic on off 여부를 판단하고 on일 경우 off로 off일 경우 on으로 전환한다.

1. **프로그램 실행방법 및 예제**

Part 0: 프로그램 시작화면

****

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

유저의 정보가 입력된 파일명을 스캔 받고 이름과 메일을 스캔해온다. 이때 총 몇 명의 user가 스캔 되었는 지를 나타내고 만들고 싶은 회의실 이름과 최대 참가자 수를 입력 받는다. 또한 다음에 실행하고 싶은 명령어를 입력 받는다. 다음 기능을 수행하기 전에 user 정보를 알파벳 순서로 정렬하도록 한다.

Part 1: 유저관리

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

저장된 유저의 이름을 알파벳 순으로 출력한다. 또한, user add를 통해 txt 파일에 저장되지 않은 유저를 입력 받고, user delete를 통해 저장된 유저 중 특정 유저를 삭제한다.

Part 2: 회의실 관리

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

회의에 추가하고 싶은 유저의 이름을 입력 받고 카메라/마이크 on off 기능을 입력 받는다. 이때 participant 리스트를 user node와 연결하여 이름, 이메일, 카메라 마이크에 대한 정보를 한번에 관리한다. 또한, conf hang up 명령을 통해 삭제하고 싶은 유저를 입력 받고 삭제를 진행한다. 마지막으로 참가자의 카메라 on off 정보를 바꾸고 싶을 때, if 문을 통해 카메라를 켜고, 끈다.

Quit 기능으로 프로그램을 종료한다.

1. **토론**

* 처음 링크 리스트를 통해 노드를 연결하는 것이 정말 어려웠다. 하지만 강의를 복습해서 듣다 보니 temp라는 방을 만들어서 연결하면 쉽다는 것을 배웠고, 실제로 사용하니 정말 쉽게 연결할 수 있었다. 삭제도 마찬가지로 lab시간에 배운 코드를 활용하니 쉽게 구현할 수 있었다.
* 재귀함수의 사용이 메모리를 차지해 에러를 일으킬 수 있다는 것을 배웠기 때문에 코드를 짜면서 재귀함수를 사용하지 않았다.
* 처음에 할당한 노드들을 어디서 해제해줘야 할지 어떻게 해제해야 할지 감이 오지 않았다. Google을 검색하며 노드를 해제하기 위해서는 임시의 방을 만들어 정보를 저장한다음 할당 해줘야 다음 노드를 찾을 수 있다는 것을 배웠고 실제로 그렇게 실행하니 정상적으로 해제되었다.

1. **결론**

* 이번 어싸인을 통해 linked list를 연결하고 삭제하는 방법에 대해 이해했다. 하지만 정말로 디테일 한 부분까지 이해한 것이 아니라, 에러 난 부분 코드를 조금씩 바꿔보다 해결된 곳이 많아서 강의를 반복해서 듣거나 교과서 예제들을 참고하면서 linked list를 구상 해야겠다고 생각했다.
* 문제 pdf에 그려진 그림을 통해 participants 리스트가 user list의 노드들을 따로따로 가리키도록 프로그래밍 할 수 있다는 것을 깨닫고 실제로 코드를 짜니 잘 실행되었다.

1. **개선 방향**

* 이번 어싸인은 마지막 어싸인이니 글자수를 줄이고 반복을 최대한 줄여보려고 노력했다. 다시 짜보라고 하면 for문으로 반복되는 모든 부분을 함수로 짜보고 싶다.
* 포인터 변수 할당, 할당 해제가 처음에 많이 고민되었다. 수업 노트와 랩시간에 배운 코드를 복습하니 할당 해제하는 지점에 대해 알게 되어 다음에 linked list를 사용한다면 문제 없을 것 같다.