**시스템 프로그래밍**



**연습문제 2.9, 2.10**

**이름: 최강석**

**학번: 2012136132**

**분반: 2**

**날짜: 2016년 10월13일**

**소스코드**

|  |
| --- |
| #include<stdio.h>  #include<fcntl.h>  #include<unistd.h>  #include<string.h>  #define NAMELENGTH 6 //이름의 길이를 5개로 할꺼니까 마지막엔 '\0'이 들어갈 수 있게  #define NROOMS 5//방을 5개로 설  char namebuf[NAMELENGTH];//버퍼 변수  int infile = -1;//파일을 읽기, 쓰기모드로 열기위해  off\_t offset; //오프셋 변수 (위치를 조절시켜주는 변수)  char \*getoccupier(int roomno)  {  ssize\_t nread;  if(infile == -1 &&  (infile = open("residents", O\_RDWR)) == -1)  //파일을 읽기, 쓰기 모드로 열어준다.  {  return (NULL);  }  offset = (roomno-1)\*NAMELENGTH;  //한줄이 넘어갈때마다 맨앞에서부터 NAMELENGTH만큼 이동하므로  //offset 설정을 (방번호 -1)\*NAMELENGTH로 해준다.  if(lseek(infile, offset,SEEK\_SET) == -1){  //커서의 위치를 처음시작에서 부터 offset만큼 이동  return (NULL);  }  if( (nread = read(infile, namebuf, NAMELENGTH)) <= 0){  //NAMELENGTH만큼 읽어 namebuf에 써준다.  return (NULL);  }  namebuf[nread - 1] = '\0'; //마지막 버퍼에 '\0'값을 넣어준다.  return namebuf;//해당하는 방의 버퍼를 반환해준다.  }  int findfree()// 가장 작은 수의 빈방을 찾는 변수  {  int i = 0;  char \*getoccupier(int), \*p;  for(i = 1; i <= NROOMS; i++ )  {//1부터 방번호까지 검사하여 만약 문자열이 EMPTY이면  //해당하는 방의 번호 출력.  if(p = getoccupier(i)){  if(!strcmp(p,"EMPTY")){  printf("The smallest and empty room number is %d \n ", i);  return 0;  }  }  }  printf("There is no empty room\n");//EMPTY가 없을 경우.  return 0;    }  int freeroom()  {  char \*getoccupier(int), \*p;  int input;  printf("Enter Remove Guest room number : ");  scanf("%d", &input);    if(p = getoccupier(input)){  if(!strcmp(p,"EMPTY")){//먼저 방이 비어있는지 확인  //방이 비어있으면 종료시켜준다.  printf("Room %d is already EMPTY ROOM\n", input);  return 0;  }  //만약에 방이 비어있지 않으면 해당하는 (방의번호-1)\*NAMELENGTH를  //이용해 offset을 설정한다. 그리고 나서 lseek으로 커서의 위치를  //설정하고 해당하는 곳에 "EMPTY\n"를 써준다.(덮어써진다.)  offset = (input-1)\*NAMELENGTH;  lseek(infile,offset, SEEK\_SET);  write(infile, "EMPTY\n", NAMELENGTH);  }  }  int addguest()  {  char \*getoccupier(int), \*p;  int input;  char Name[NAMELENGTH];  printf("Enter add Guest room number : ");  scanf("%d", &input);  //비었는지 먼저 확인을합니다.  if(p = getoccupier(input)){  if(!strcmp(p,"EMPTY")){//만약에 비어있다면  printf("Room %d is EMPTY ROOM\n", input);  printf("NewGuest Name : ");//새로운 게스트를 입력받습니다.  scanf("%s", Name);  Name[NAMELENGTH-1]='\n';//입력을 받고 마지막 문자열을 '\n'으로 설정  // (방의번호-1)\*NAMELENGTH를 이용해 offset을 설정한다.  // 그리고 나서 lseek으로 커서의 위치를  //설정하고 해당하는 곳에 Name 써준다.(덮어써진다.)  offset = (input-1)\*NAMELENGTH;  lseek(infile,offset, SEEK\_SET);  write(infile, Name, NAMELENGTH);  }  else //빈방이 아닐경우에 출력.  printf("Room is not EMPTY\n");  }  }  int main(){    int j, option;  char \*getoccupier(int), \*p;    while(1){  //옵션을 선택할 수 있게 합니다.  printf("\n---Select option---\n");  printf("1------Room Status\n");//방의 상태  printf("2---------finefree\n");//빈방 중 가작 작은 번호의 방  printf("3---------freeroom\n");//빈방을 만들어줍니다.  printf("4---------addguest\n");//방에 게스트를 더합니다.  printf("5-------------QUIT\n");//종료합니다.  scanf("%d", &option);  switch(option){  case 1://1번을 누를 경우 방의 현재 상태를 나타냅니다.  {  for( j = 1; j <= NROOMS; j++)  {  if(p = getoccupier(j))  printf("Room %2d, %s\n", j, p);  else  printf("Error on room %d\n", j);  }  break;  }  case 2://2번을 누를 경우 빈방 중  //가장 작은 번호의 방이 출력됩니다.  {  findfree(); break;  }  case 3://방을 비움니다.  {  freeroom(); break;  }  case 4://방에 게스트를 더합니다.  {  addguest(); break;  }  case 5: //종료합니다.  return 0;  default : return 0;  }  }  close(infile);  } |

**설명**

먼저 switch문을 이용하여서 연습문제 2.9와 2.10에서 요구하는 것들을 출력을 하려고 하였습니다. 2.9에서는 방이 비었는가를 결정하는 방법을 고안하라고 했다. 이를 어떻게 할지 생각해 봤는데 교수님께서 짧게5글자를 투숙객의 정보로 사용하라고 하신 것이 생각나서 빈방에는 ‘EMPTY’라는 글

자를 넣으면 손쉽게 해결할 수 있을 것 같았습니다. 먼저 getoccupier같은 경우는 수업내용이나 책에 코드가 다 있어서 문제가 없었습니다. 빈 반 중 방 번호가 가장 작은 방을 찾은 findfree루틴은 1번 방부터 검사해서 EMPTY(빈방)이 나오면 그방을 출력 시켜주고 함수를 종료시키게 해주었습니

다. 먼저 EMPTY검사는 string.h의 strcmp를 이용했습니다.

다음은 freeroom루틴입니다. 이는 파일 입출력에서 파일에 쓰기가 필요해서 처음엔 파일 오픈할 때 옵션으로 O\_RDONLY를 하였지만 이후 O\_RDWR로 바꾸어 주었습니다. 처음에 지우고 싶은 빈방의 번호를 입력 받습니다. 그 이후로는 먼저 방이 비어있는지 확을하고 방이 비어있으면 종료시켜 줍니다. 이때도 EMPTY검사를 strcmp로 해주었습니다. 만약에 방이 비어있지 않으면 해당하는 (방의번호-1)\*NAMELENGTH로 lseek으로 offset을 설정합니다. 설

정하고 해당하는 곳에 write로 “EMPTY\n”를 써줍니다. 다음은 addguest

입니다. 먼저 그자리가 비어있는지를 확인합니다. 비어있을 경우 새로운 게스트를 받습니다. 다음은 freeroom과 마찬가지로 offset 과 lseek을 이용해

서 해당문자열로 이동을 시켜줍니다. 다음은 write로 사용자부터 입력된 na

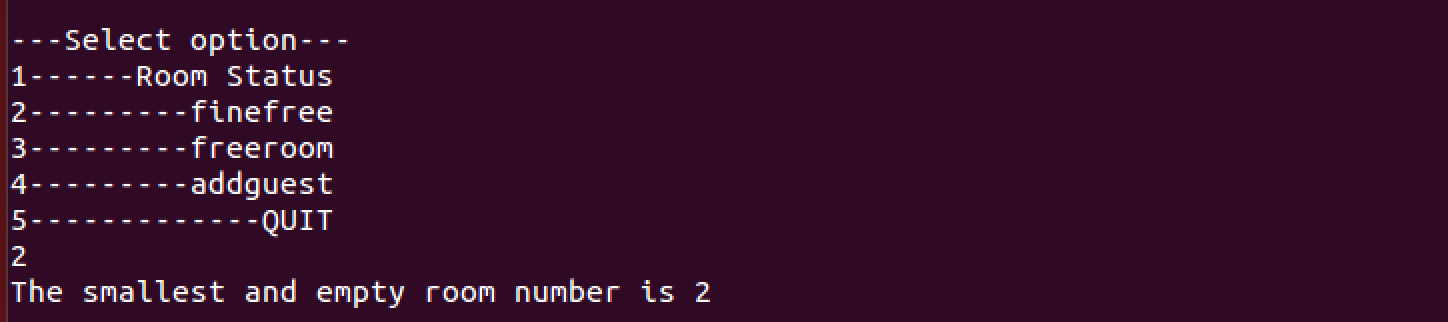
me을 파일에 출력할 수 있게 해줍니다.

**실행화면**



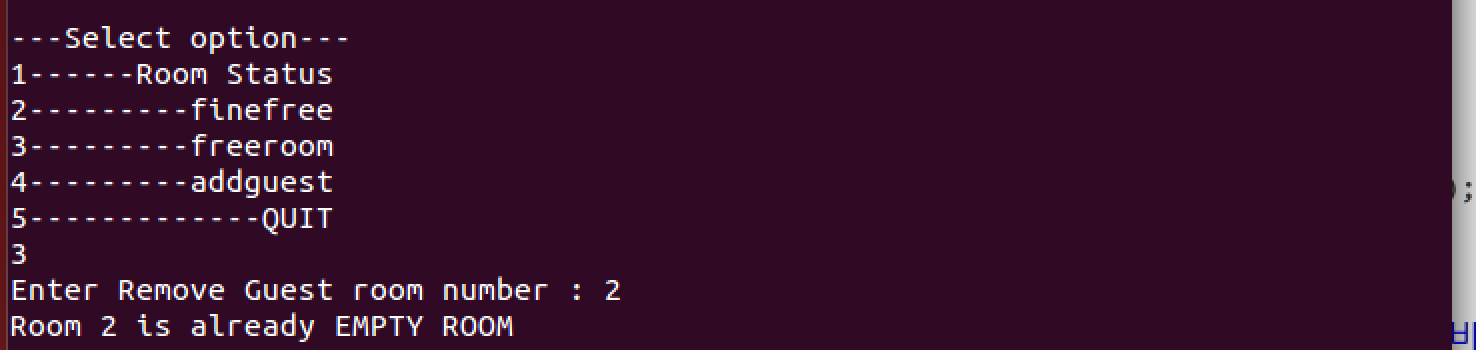
처음화면에서 Select option에서 Room status를 해준 경우. 해당하는 방에

대한 투숙객이 나온다. EMPTY는 현재 빈 방인 상태입니다.



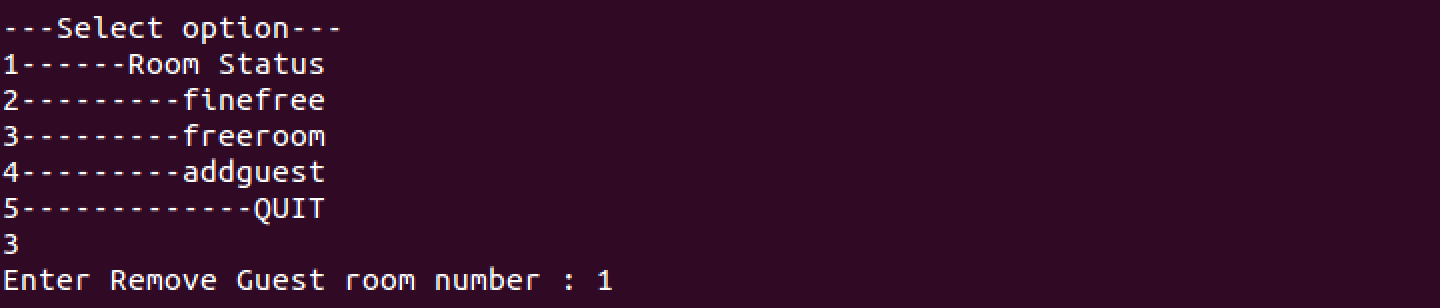
다음은 2번 finefree를 실행 시켰을 경우 가장 번호가 작은 빈방은 2번 방이

나온다 위에서 보기와 같이 2번방이 빈방 중 가장 작은 방입니다.



3번을 입력했을 경우 2번 방을 비우라고 입력을 했다. 하지만 2번방은 현재

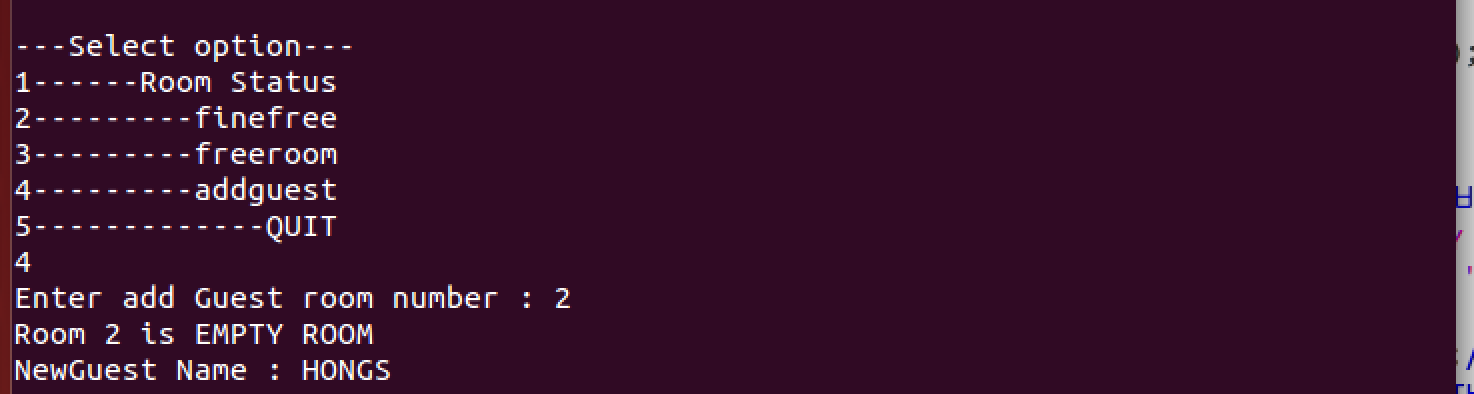
빈방이기 때문에 이미 빈방이라는 말을 띄웠습니다.



다음은 3번을 입력하고 투숙객이 있는 1번 방을 비우라고 했습니다.



Status결과는 다음과 같이 1번방이 EMPTY로 바뀌었습니다.

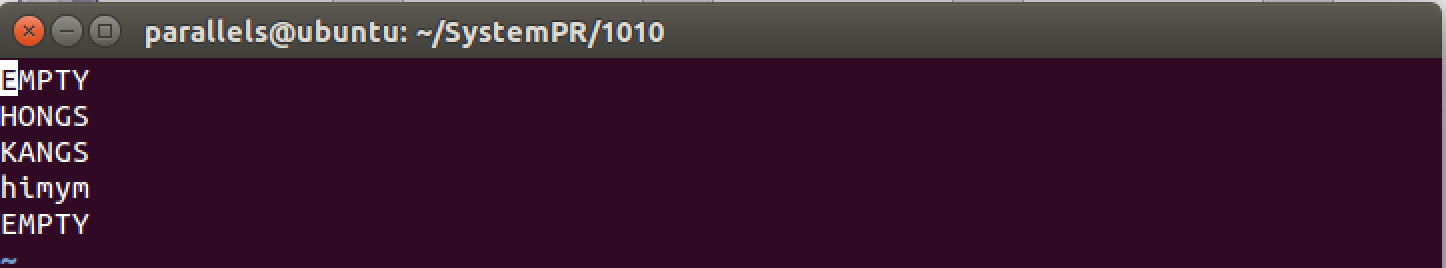


다음은 4번(addguest)를 입력하였습니다. 먼저 인원을 넣고싶은 방의 번호 2

번을 입력합니다. 그럼 방이 빈지 안빈지를 확인하고 나서 새로운 투숙객의 이름을 입력을 합니다.



2번에 입력한 HONGS가 들어왔습니다.



vi로 residents를 열어본 결과 문제 없이 나왔습니다.