컴퓨터 구조 pa0.c REPORT

202126469 홍리나

1. 기본적인 logic

- main에서 파일 입출력으로 한줄씩 읽어오는 것을 확인
- 반복문을 통해 한 줄 읽어옴
  - 읽다가, 공백이 아닌 문자를 만나면 tokenArray 배열에 저장
    - token의 길이를 len_tok에 별도 저장
  - len_tok가 0보다 크고, tokenArray[0]이 문자열의 끝이 아니라면 -> tokenArray에 null이 들어가서 len_tok>0을 만족해 빈 문자열이 token에 대입 되는 경우 방지
  - token의 길이만큼 동적 할당을 한 tokens[]에 한글자씩 대입한 후, 마지막이 문자열의 끝임을 알리기 위해 \0 대입
- 반복문 종료시, token의 갯수를 담은 tok_ctr을 nr_tokens에 대입
- 이 함수가 종료되면, 한줄의 문자열에서 띄어쓰기를 기준으로 단어를 추출하여 tokens에 저장

2. Pa0.c code

```
static int parse_command(char *command, int *nr_tokens, char *tokens[])
{
    int tok_ctr = 0;
    for(int ctr = 0; command[ctr] != '\0'; ctr++){ // 한줄씩 읽어옴
        char tokenArray[64]={0,};
        int len_tok = 0;

        for (int i=0; isspace(command[ctr]) == 0; i++) {  // 공백이 아닌 경우
            tokenArray[i] = command[ctr];
            ctr++;
            len_tok = i+1;
        }

        if(len_tok>0 && tokenArray[0] != '\0'){ // 토큰이 존재하는 경우, token 대입
            tokens[tok_ctr] = (char*)malloc(sizeof(char)*(len_tok+1)); // 토큰 길이만큼 메모리 할당
            for(int i=0; i<len_tok; i++){ // 토큰을 tokenArray에서 tokens로 복사
                tokens[tok_ctr][i] = tokenArray[i];
            }
            tokens[tok_ctr][len_tok] = '\0'; // 토큰의 끝에 NULL 대입
            tok_ctr++;
        }
    }
    *nr_tokens = tok_ctr;
    return 0;
}
```