2016312761 assignment#1 구현 방법과 디자인

전체적인 convert.c파일의 구조는 convert함수와 strcat, strlen, strcpy 함수로 이루어져 있습니다. #0과제에서 구현해보았던 함수를 바탕으로 string.h헤더파일을 쓰지 않고 활용해보았습니다.

우선 인풋파일을 처음부터 끝까지 읽어서 buf라는 char배열에 한꺼번에 넣었습니다. 배열에 넣어놓아야 여러가지 연산, 처리가 가능하기 때문입니다. 그리고 for문을 통해 buf배열을 한 칸 씩 확인하면서 \n의 개수와 \r\n의 개수를 세었습니다. 각각 ncnt와 rncnt라는 변수에 넣은 뒤 비교하여 unix로 바꿀지 dos로 바꿀지를 결정하였습니다.

Dos로 바꿔야 할 경우에, for문으로 buf배열을 한 칸 씩 확인하면서 outputfile에 1개씩 write해줍니다 여기서, 어떤 칸에 \n이 들어있는데 그 앞에 칸이 \r이 아니라면, unix포맷이기 때문에 \r을 한 번 write하고 확인하던 칸에 들어있는 것도 write해줍니다. 또한, \t가 오면, \t대신에 quadspace라는 변수에 저장되어 있는 space4칸 “ “를 write 해줍니다. Unix로 바꿔야 할 경우에는 특별한 조건이 더 붙는데, 인풋파일이 .c, .h, .java중에 하나라면, 연속된 space8개를 tab으로 바꿔줘야 합니다. 그래서 처음에 인풋파일이름을 받고 strlen을 사용하여 if문을 통해 .c, .h, .java중 하나라면 chjava라는 int형 변수를 0에서 1로 바꿔주는 걸로 구현하였습니다. Dos로 바꾸는 것과 비슷하게, \r이 왔는데 다음 칸이 \n이라면 \n만 write에서 받아주고 넘어가고, 나머지는 전부 write해주었습니다. .c.h.java일 경우에 space가 오기 시작하면 그 개수만 세고 write는 하지 않다가, 8개가 되기 전에 다른 문자가 왔다면, for문을 통해 개수만큼 space를 write하고 다른 문자를 write한 뒤 그 개수를 가리키는 변수는 0으로 초기화합니다. 만약 8개가 되었다면, \t를 써주고 변수는 0으로 초기화합니다. 이것으로 write의 과정이 전부 끝나면, 프린트 양식에 따라 인풋파일이름, 아웃풋파일이름, 아웃풋파일의 바이트 개수를 출력하고 파일을 닫고 종료됩니다.

옵셔널 기능을 구현하는 코드는 Dos로 바꿀 때나 Unix로 바꿀 때나 모두 동일합니다. for문으로 buf를 한 칸 씩 확인하기 전에, for문 안 첫 부분에서 while문을 통해 스페이스바나 탭이 있다면, 안 나올 때까지 계속해서 다음 칸으로 넘어가고, 넘어간 칸의 개수를 trashcnt라는 변수에 저장합니다. 그래서 while문을 빠져나왔을 때, 그 문자가 엔터라면 for문 반복 횟수에 관련된 i변수에 trashcnt만큼 더해줘서 쓸데없는 칸들이 write하는 코드에 관여되지 않도록 합니다. 만약 엔터가 아니라 다른 문자라면 그냥 write진행하면 됩니다.