데이터 구조

단어장 프로그램 프로그래밍

컴퓨터 공학

2015211373 권정인

목차

1. 개요 및 작업 환경
2. 전체 코드 및 코드의 요약
3. 컴파일 및 결과
4. 평가 및 개선 방안
5. 개요 및 작업 환경

C언어 코드 작성 시작 전, 유닉스의 SCP(Secure Copy) 명령을 통해 우선 학교 서버 내에 있는 /usr/dict/words 파일을 제 개인 컴퓨터로 복사한 후 시작하였습니다. 이에 대한 명령어는 다음과 같습니다 : *scp –P 11800 [학과서버 ID]@dalma.dongguk.ac.kr:/usr/dict/words ~/*

저는 Windows 10 내에 있는 Ubuntu 16.04 기반 Linux 서브 시스템 내에서, 코드 작성은 VI로, 코드 빌드는 GCC로 실행 하였습니다.

코드 실행을 제 로컬 장치가 아닌, 교내 유닉스 시스템 내에서 실행한다면, 소스 내의

*FILE \*fp=fopen("words", "rt");* 을

*FILE \*fp=fopen("/usr/dict/words", "rt");* 로 수정하면 됩니다.

1. 전체 코드 및 코드의 요약

#include <stdio.h>

#define MAX 25150 *//기호 상수를 이용하여 전체 단어 수 MAX 정의*

int main(void)

{

char buf[100]; *//단어 한개의 길이를 최대 100자까지 표현하도록 정의합니다.*

int i,j; *//for를 이용하기 위한 변수 선언*

int state; *//후에, fclose를 이용하여 단어가 정상적으로 닫혔는지에 대한 여부 확인*

int num = 1; *//단어장 앱을 실행할 때, 단어 번호를 표기합니다.*

FILE \*fp=fopen("words", "rt"); *//개인 홈 디렉토리에 있는 words 파일을 텍스트 읽기 모드로 열기*

if(fp==NULL){

printf("not open");

return 1; *//파일이 열리지 않았을 경우 fp는 NULL로 정의되며, 파일을 열 수 없다는 문구와 함께 return 1로 프로그램을 종료합니다.*

}

for(j=0;j<=(MAX/25);j++){ *//전체 단어 수 MAX에 25를 나눔을 이용하여 페이지 수 전환*

system("clear"); *//가독성 향상을 위해 UNIX 명령어인 ‘clear’에 접근합니다.*

printf("\t---Words---\n");

for(i=0; i<25; i++){ *//한 페이지에 단어를 하나씩 25번, 총 25개의 단어를 출력합니다.*

if(fgets(buf, sizeof(buf), fp)==NULL) break; *//fgets를 이용하여 단어장을 가져왔을때 값이 NULL을 선언하면 break합니다.*

printf("%5d %s",num,buf); *//num은 단어 번호, buf은 각 단어 내용*

num++; *//단어 하나를 출력한 다음 num을 1씩 더하여, 단어 번호를 증가시킵니다.*

}

printf("\tPages : %d\n",j+1); *//페이지의 첫 값 J가 0이므로 1을 더하여 1페이지부터 시작하도록 설정*

printf("Press [Enter] to show next words\n");

getchar(); *//[Enter] 버튼을 눌러 다음 페이지를 출력하세요.*

system("clear"); *//다음 페이지를 로드하기 전, 가독성 향상을 위해 UNIX의 clear 명령에 직접 접근합니다.*

}

state = fclose(fp); *//파일이 정상적으로 닫혔는가에 대한 여부를 state로 정의*

if(state!=0){

printf("We couldn’t close file.");

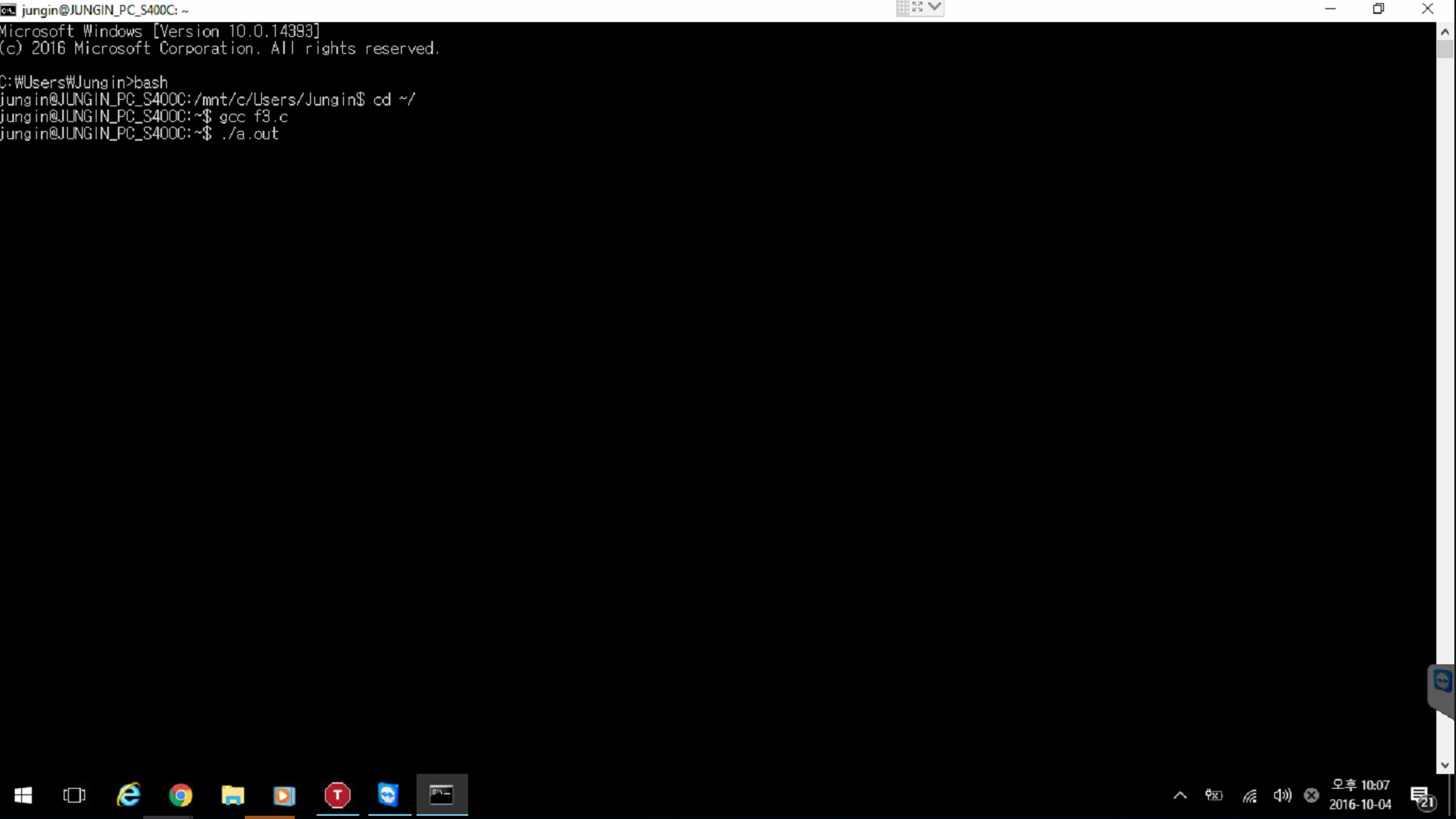
return 1; *//state가 0(정상)이 아닐 경우, 오류 메시지 출력과 함께 return 1을 반환합니다.*

}

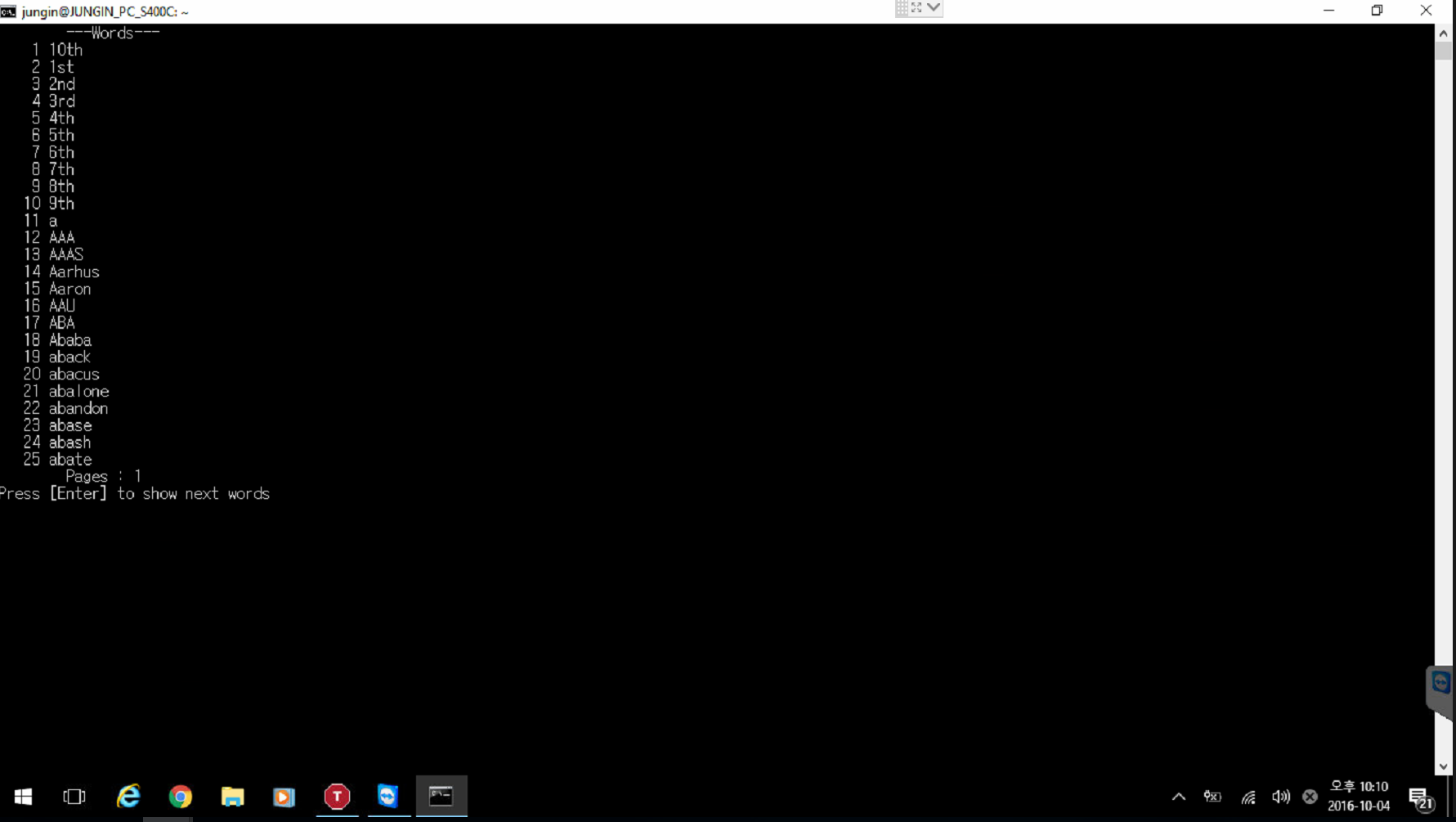
return 0; *//모든 작업을 마친 후, return 0을 반환하여 정상적으로 프로그램을 종료하세요.*

}

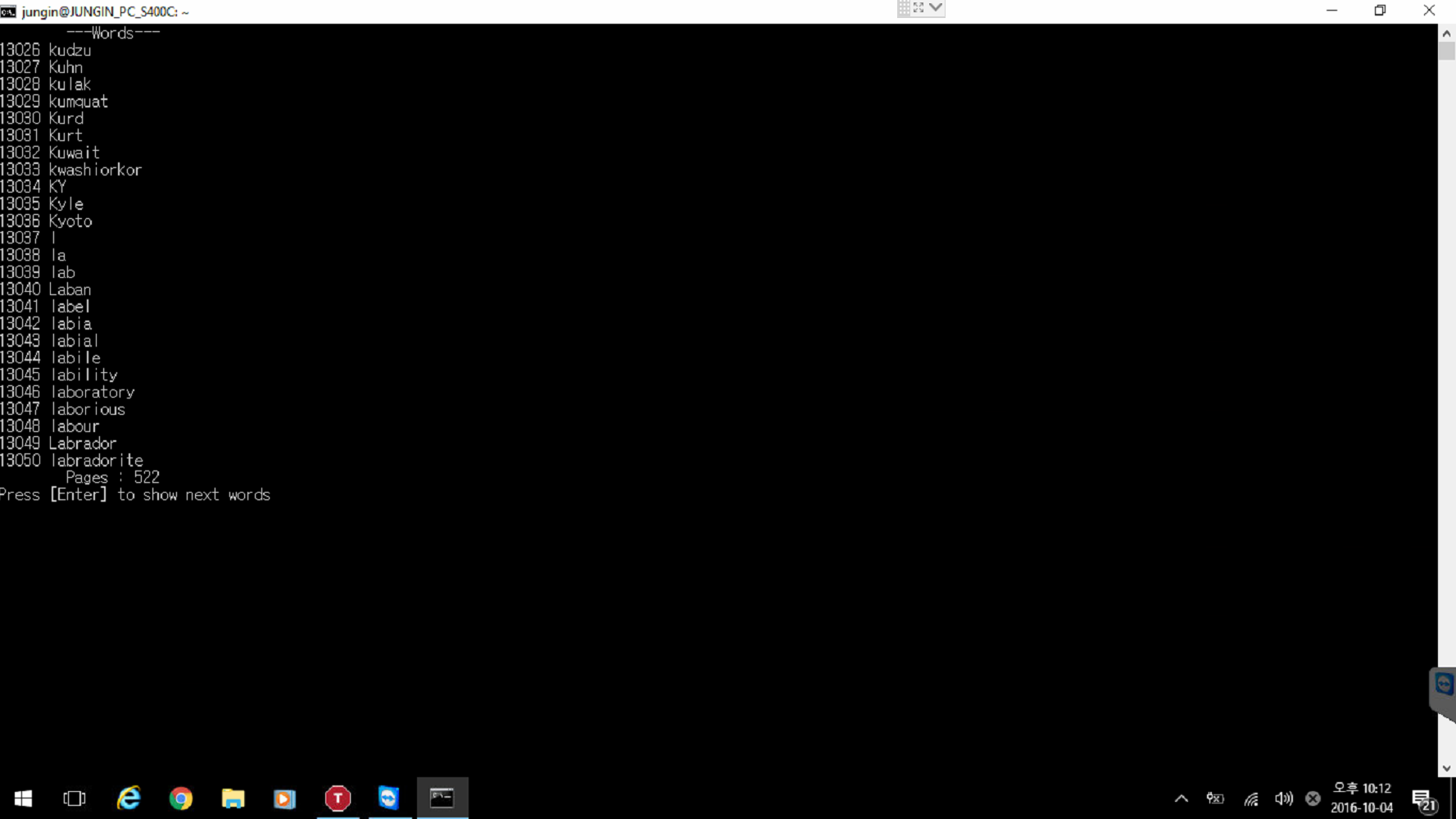
1. 컴파일 및 결과



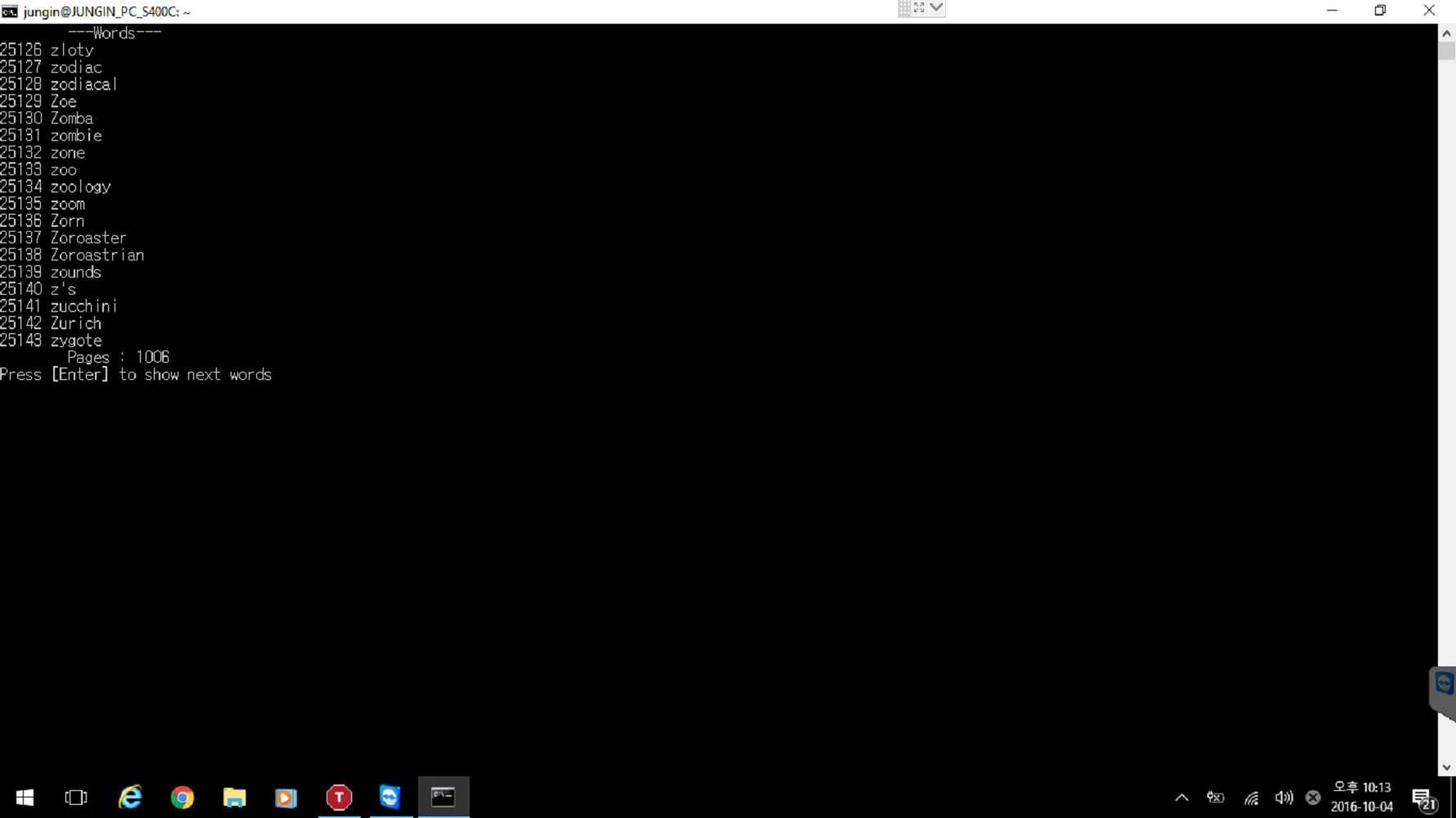
1. Windows 10 내의 bash 서브시스템으로 접근 후, 개인 홈 디렉터리로 진입하고, gcc를 이용해 소스 파일 f3.c를 컴파일하고, 컴파일하여 출력 된 a.out 프로그램을 실행합니다.



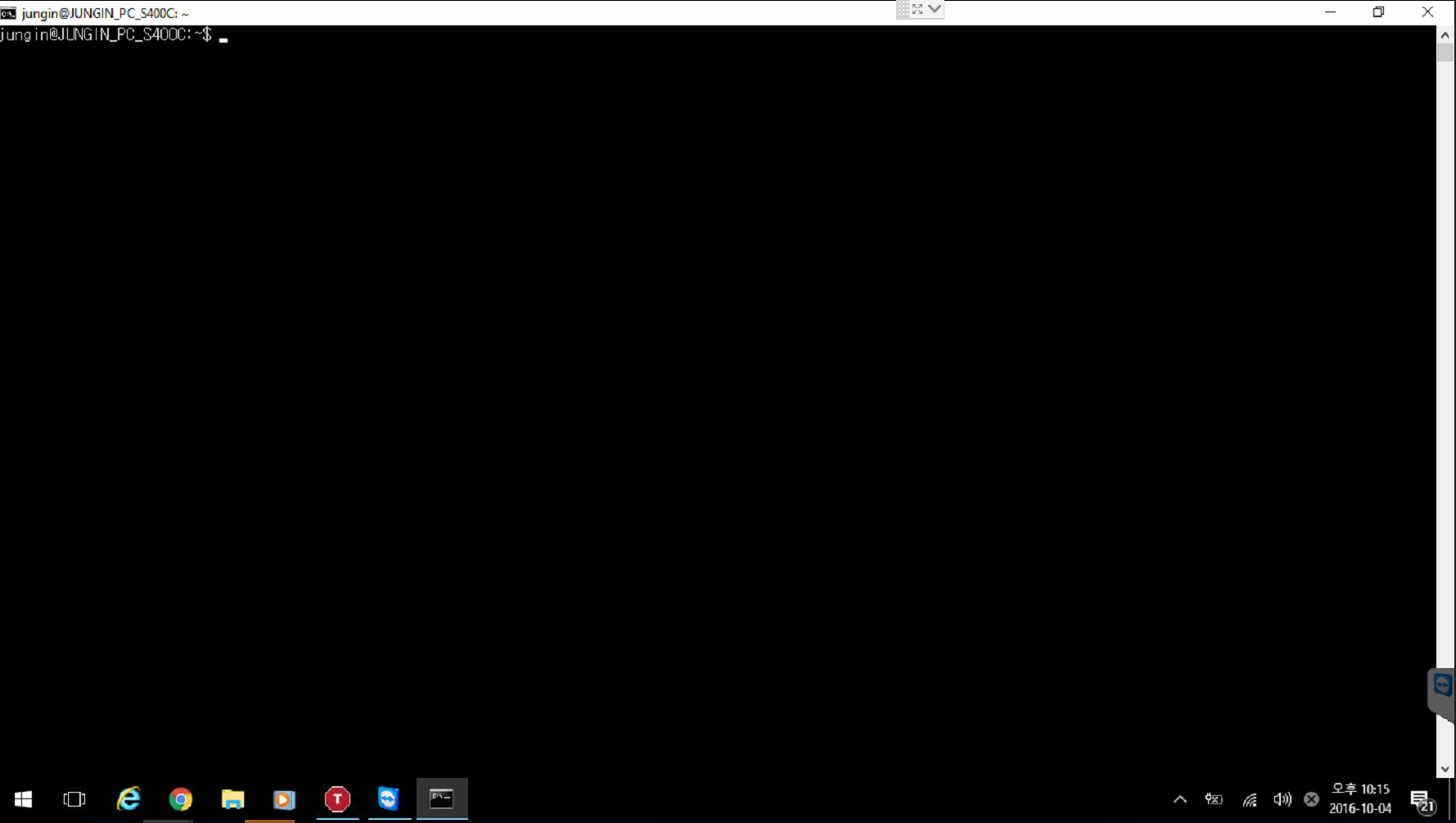
2. [a.out] 프로그램을 실행한 첫 화면



3. [a.out]파일을 실행한 중간 지점 화면



4. [a.out] 프로그램을 실행한 후, 단어의 마지막을 출력하는 페이지가 출력되는 화면



5. 마지막 부분이 출력된 이후, 다시 한번 [Enter]를 누르면, 화면이 clear되며, return 0을 반환함으로서 프로그램이 정상적으로 종료됩니다

1. 평가 및 개선 방안
2. 코드를 작성하는 과정에서, 화면을 비우는것을 프로그램 내에서 처리하려는데 있어, 어려움이 닥쳤습니다. 저는 이것을 해결하기 위해, 프로그램 내에서 유닉스 시스템에 직접 명령어를 실행시키는 system() 을 사용하였습니다.
3. 저는 개인적으로, 소스 파일 작업 및 실행을 교내 서버에 SSH로 연결하지 않고, 로컬 장치에서 실행하고 싶었습니다. 하지만 교내 서버에 있는 words를 가져오는 방법을 알지 못하였습니다. 그래서 인터넷을 검색하는 도중, 원격 서버 내 파일을 로컬 장치로 가져오는 scp 라는 명령어를 발견하였고, 이를 이용하여 교내 서버 내의 파일을 제 로컬 장치로 가져와 작업하였습니다.
4. 개인적으로 C언어를 완벽하게 구현하지 못하여, 몇몇 코드는 인터넷을 검색하여 참조하여 작성하였습니다.
5. 코드를 보다 더 간략화하고 효율적으로 작성하여, 프로그램 실행 시 시스템 리소스를 조금 적게 차지하도록 하여야 할 것 같습니다.
6. 프로그램 내에서 if 문을 활용하여 마지막 페이지에 도달하였을 경우,

*Press [Enter] to show next words.* 가 아닌

*Press [Enter] to exit program.* 이 출력되도록 구현한다면, 조금 더 사용하기 편한 프로그램을 만들 수 있을 것 같습니다.