**System Programming**

**#2**

1. **Directory Scan**

**1분반**

**장경식 교수님**

**2012136020 김동우**

**2016.10.09**

1. **교재의 PrintDir.c 예제 분석**

#include <unistd.h>

#include <stdio.h>

#include <dirent.h>

#include <string.h>

#include <sys/stat.h>

#include <stdlib.h>

void printdir(char \*dir, int depth) {

DIR \*dp;

struct dirent \*entry;

struct stat statbuf;

/\* opendir함수는 dir파일 안에 있는 entry를 읽는다. 순차적으로 읽은 후 끝까지 다 읽으면 NULL을 반환한다. \*/

if((dp == opendir(dir)) == NULL) {

fprintf(stderr, “cannot open directory: %s\n”, dir);

return;

}

chdir(dir);

/\* readdir함수는 dir항목을 가리키는 포인터를 반환한다. 호출을 성공하면, 다음 dir를 반환하고, 끝을 만나면 NULL을 반환한다.

Entry에 저장된 dir명을 상태버퍼에 저장 한다.

\*/

while((entry = readdir(dp)) != NULL) {

lstat(entry->d\_name, &statbuf);

//파일의 유형이 dir형식인지 확인한다.

if(S\_ISDIR(statbuf.st\_mode)) {

//dir을 찾아, “.”와 “..”을 무시한다.

//”.”혹은 “..”이면 다음 계속해서 엔트리를 읽는다.

if(strcmp(“.“,entry->d\_name) == 0 ||

strcmp(“..“,entry->d\_name) == 0)

continue;

printf(“%\*s%s/\n“,depth,““.entry->d\_name);

//printdir함수를 재귀적으로 호출

printdir(entry->d\_name, depth+4);

}

//dir가 아닌 파일일 경우, 파일명을 출력한다.

else printf(“%\*s%s\n“,depth,““,entry->d\_name);

}

// 열었던 dir를 닫는다.

chdir(“..“);

closedir(dp);

}

int main() {

printf(“Directory scan of /home:\n“);

printdir(“/home“,0);

printf(“done. \n“);

exit(0);

}

1. **Directory Scan**

- hidden 파일 (파일명이 dot(.)로 시작하는 파일)은 제외할 것.

- 현재 디렉토리의 하부 디렉토리(sub-directory)도 재귀적으로 탐색한다. 즉, 현재

- 디렉토리에에 속한 디렉토리 내용도 같은 방식으로 출력한다.

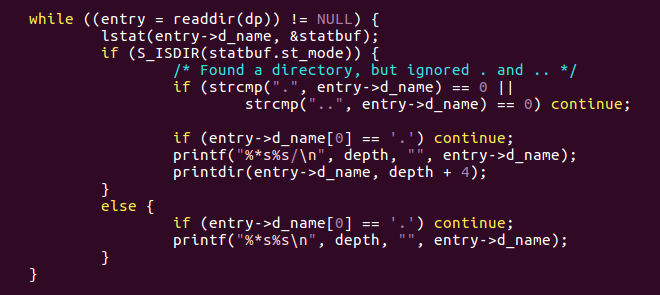
- 하부 디렉토리 구분은 tab문자를 추가한다.

1. 문제 분석

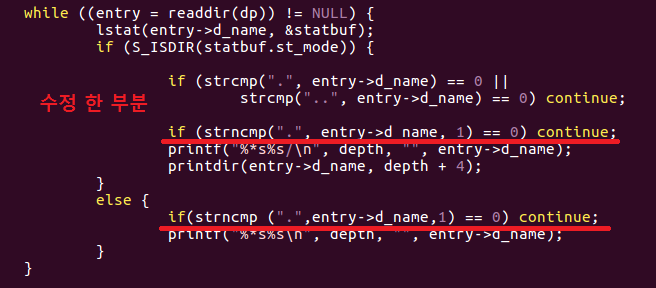
문자열을 비교하는 함수 중 지정한 개수만 비교하는 함수인 strncmp를 사용한다. strncmp(".", entry -> d\_name, 1)로 설정하면 파일 이름 중 첫 번째 문자를 비교하여 . 일 경우는 0, 아닐 경우는 1로 반환된다. 이를 이용하여 이름 중에 . 으로 시작하는 파일은 출력하지 않도록 하는 if문을 작성한다.

If(strncmp(“.”, entry -> d\_name,1) == 0) continue;

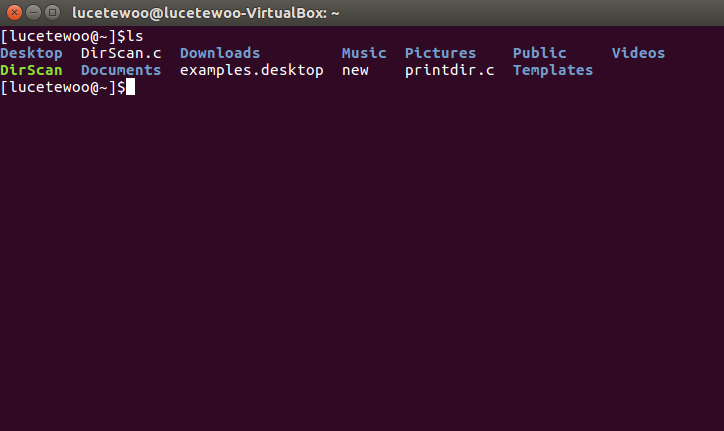
**<DirScan 파일 수정 전>**



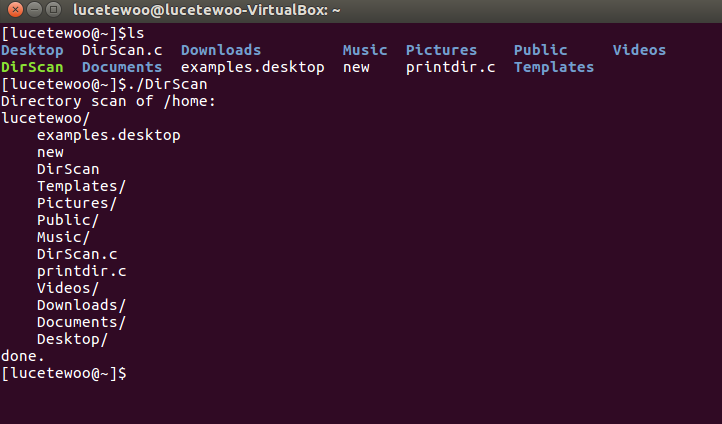
**<DirScan 파일 수정 후>**



**<DirScan 컴파일 후 실행파일 생성>**



**<DirScan 파일 실행>**



**실행 된 파일들을 보면 dir에 대해서는 tab문자를 추가하여 출력되었고, 파일명이 dot(.)로 시작하는 hidden파일도 제외하고 출력된 것을 알 수 있다.**

1. **느낀점**

이번 과제는 디렉토리의 내부 파일들을 출력하는 dirScan파일을 작성하고, 화면에 출력하는 것이었다. 처음에는 수업 자료에 제시 되어 있는 예제 파일을 보면서 소스들을 분석하고, 이해하였다. 한없이 어려울 것만 같았던 시스템프로그래밍에 대해 직접 코드를 수정해보고, 실행하여 결과물을 보니 조금씩 흥미가 생기는 것 같다. 이제 vi를 이용하여 파일을 편집하고, gcc를 활용하여 컴파일 하는 방법에는 수월해 진 것 같다. 앞으로의 주어진 과제들에 대해서도 많은 것을 배우고 시스템 프로그래밍에 대해 한발 더 나아가고 싶다.