컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <dirent.h>

#include <limits.h>

#include <string.h>

#include <sys/stat.h>

#include "ssu\_runtime.h"

#ifdef PATH\_MAX // PATH\_MAX가 정의되어 있다면

static int pathmax = PATH\_MAX; // pathmax에 대입

#else // 아니라면

static int pathmax = 0; // 0을 대입

#endif // if문 종료

#define MAX\_PATH\_GUESSED 1024

#ifndef LINE\_MAX

#define LINE\_MAX 2048 // LINE\_MAX를 정의

#endif

char \*pathname;

char command[LINE\_MAX], grep\_cmd[LINE\_MAX];

int ssu\_do\_grep(void)

{

struct dirent \*dirp;

struct stat statbuf;

char \*ptr;

DIR \*dp;

if (lstat(pathname, &statbuf) < 0) { // 파일을 lstat() 본래 파일의 stat구조체를 저장

fprintf(stderr, "lstat error for %s\n", pathname);

return 0;

}

if (S\_ISDIR(statbuf.st\_mode) == 0) { // 파일이 디렉토리 파일이 아니라면

sprintf(command, "%s %s", grep\_cmd, pathname); // command에 명령어를 넣고

printf("%s : \n", pathname); // 폴더명 출력 후

system(command); // 명령어 실행

return 0; // 종료

}

ptr = pathname + strlen(pathname); // 포인터에 문자열의 마지막의 NULL을 가르키고있는 주소값 대입

\*ptr++ = '/'; // 그자리에 ‘/’를 넣고

\*ptr = '\0'; // 다음자리에 NULL을 넣는다

if ((dp = opendir(pathname)) == NULL) { // 폴더를 opendir()

fprintf(stderr, "opendir error for %s\n", pathname);

return 0;

}

while ((dirp = readdir(dp)) != NULL) { // 폴더의 모든 파일을 읽을 떄 까지

if (strcmp(dirp->d\_name, ".") && strcmp(dirp->d\_name, "..")) { // . 과 .. 이 아닐때만 (재귀가 무한으로

strcpy(ptr, dirp->d\_name); // 읽은 파일의 이름을 복사 // 깊어지는 것을 방지

if (ssu\_do\_grep() < 0) { // 재귀적으로 돌면서 실행 에러 시

break; // 반복문 탈출

}

}

}

ptr[-1] = 0; // 위에서 NULL을 넣었던 자리 앞칸(즉, ‘/’의 자리)에 0 대입

closedir(dp); // closedir()

return 0;

}

void ssu\_make\_grep(int argc, char \*argv[]) {

int i;

strcpy(grep\_cmd, " grep"); // grep\_cmd에 “ grep”을 복사

for (i = 1; i < argc - 1; i++) {

strcat(grep\_cmd, " "); // 한칸 띄고

strcat(grep\_cmd, argv[i]); // 인자로 받은 폴더명을 붙여 쓰기

}

}

int main(int argc, char \*argv[])

{

gettimeofday(&begin\_t, NULL);

if (argc < 2) {

fprintf(stderr, "usage : %s <-CVbchilnsvwx> <-num> <-A num> <-B num> <-f file> \n"

" <-e> expr <directory>\n", argv[0]);

exit(1);

}

if (pathmax == 0) { // pathmax가 0이면 (PATH\_MAX가 정의 되어있지않을때)

if ((pathmax = pathconf("/", \_PC\_PATH\_MAX)) < 0) { // 새로운 값을 지정

pathmax = MAX\_PATH\_GUESSED;

}

else {

pathmax++;

}

}

if ((pathname = (char \*) malloc(pathmax + 1)) == NULL) { // pathname에 메모리를 할당

fprintf(stderr, "malloc error\n");

exit(1);

}

strcpy(pathname, argv[argc - 1]); // pathname에 인자로 받은 마지막 인자를 복사

ssu\_make\_grep(argc, argv); // 명령어를 만들고

ssu\_do\_grep(); // 파일안에 원하는 keyword가 들어있는지 탐색하는

// 함수 실행

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

1. 실행 결과

