컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include "ssu\_runtime.h"

#define BUFFER\_SIZE 1024

void ssu\_setbuf(FILE \*fp, char \*buf);

int main(void)

{

char buf[BUFFER\_SIZE];

char \*fname = "/dev/ttys004"; // 출력을 해줄 터미널의 이름

FILE \*fp;

gettimeofday(&begin\_t, NULL);

if ((fp = fopen(fname, "w")) == NULL) {

fprintf(stderr, "fopen error for %s\n", fname); // 출력을 해줄 터미널을 fopen()

exit(1);

}

ssu\_setbuf(fp, buf); // 파일 포인터의 버퍼링을 buf로 바꿔준다

fprintf(fp, "Hello, ");

sleep(1);

fprintf(fp, "UNIX!!");

sleep(1);

fprintf(fp, "\n");

sleep(1); // 터미널에 모든 단어가 개행을 만나 한번에 출력

ssu\_setbuf(fp, NULL); // 터미널의 버퍼링을 NULL로 바꿔준다

fprintf(fp, "HOW");

sleep(1);

fprintf(fp, " ARE");

sleep(1);

fprintf(fp, " YOU?");

sleep(1);

fprintf(fp, "\n"); // 터미널에 한 단어씩 출력

sleep(1);

gettimeofday(&end\_t, NULL);

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

exit(0);

}

void ssu\_setbuf(FILE \*fp, char \*buf)

{

size\_t size;

int fd;

int mode;

fd = fileno(fp); // 파일포인터의 디스크립터를 받아온다

if (isatty(fd)) { // 현재 주어진 터미널의 아이디를 가져와 해당 장치의 버퍼를 조작

mode = \_IOLBF; // 제대로 열렸다면 라인-버퍼 모드를 모드에 저장

}

else {

mode = \_IONBF; // 아니라면 논-버퍼 저장

}

if (buf == NULL) {

mode = \_IONBF; // buf 가 NULL이라면 논버퍼 저장

size = 0; // 논-버퍼여서 사이즈를 0으로

}

else {

size = BUFFER\_SIZE; // 아니라면 정해둔 사이즈를 저장

}

setvbuf(fp, buf, mode, size); // 터미널의 버퍼를 저장해둔 모드와 사이즈로 설정

}

1. 실행 결과

