

## Programming Assignment #1

**Due: September 22, 11:59:59 PM**

### 1. Introduction

- 본 과제를 통해 시스템 콜을 이용하여 파일 입출력을 수행하게 된다. 이번 과제의 목적은 커널 시스템 콜을 이용한 저수준 파일 입출력과 리눅스 프로그래밍 환경에 친숙해지는 것이다.

### 2. Overview

- 주어진 어떤 텍스트 파일을 읽고, 내용을 표준 출력으로 내보내거나, 다른 파일을 만들어 원본을 복사해 저장한다.

### 3. Specification

- 이번 과제를 수행하기 위해 구현해야 할 함수는 다음과 같다.

```
int parse_url(char *input_url, char *output_path);
```

- 이 함수는 `input_url`로 주어진 파일을 읽어 `output_path` 파일로 파일을 복사하는 함수이다.
- Arguments
  - ✓ 첫 번째 인자 `input_url`는 URL 형식으로 이루어진 파일 주소이다.
    - ◆ URL 형식 파일 주소는 파일 프로토콜이며, "file:// + localhost(생략가능) + '/' + 상대경로"의 형태로 구성된다. 예를 들어, 다음 두 경우는 모두 유효하다.  
`file://localhost/test/data.txt` or `file:///test/data.txt`
  - ✓ 두 번째 인자 `output_path`는 `input_url` 파일이 복사되어 저장될 파일의 경로 명이고, NULL이 들어올 수 있다.

- 입출력 타입의 정의

- ✓ `input_url`로 입력 받는 파일 프로토콜은 다음 형식으로 해석한다.
  - ◆ `file://localhost/test/data.txt` 혹은 `file:///test/data.txt` 문자열을 입력받았을 때, `$(HOME_DIR)/test/data.txt` 파일을 읽어야 한다.
    - `$(HOME_DIR)`은 환경 변수(environment variable) `HOME_DIR`을 나타내는 표현이며, 환경 변수는 C 표준 함수 `char* getenv(const char* name)` 함수로 값을 알 수 있다.
    - `$(HOME_DIR)`의 서버상 경로는 채점 서버에서 임의로 주어진다. 즉, `getenv()` 함수의 결과물과 파일 프로토콜을 분석하면, `input_url`가 지정하는 파일의 서버상 절대 경로를 만들 수 있다.
- ✓ `output_path`로 입력 받는 문자열은 그대로 `open()`의 인자로 입력하여 파일 출력한다. 단, `output_path`가 `NULL`로 주어진 경우, 출력은 표준 출력에 수행한다.
  - ◆ `output_path`에는 파일이 존재하지 않는다고 가정한다. 이미 존재하는 파일을 다루는 방법을 고려할 필요가 없다.

- 결과의 반환

- ✓ `parse_url()` 함수를 통해 정상적으로 파일을 읽고 처리한 경우 해당 파일의 크기를 반환한다.
  - ◆ 파일의 크기는 `read()` 한 양을 합산하거나, `stat()` 함수를 통해 알 수 있다.
  - ◆ 에러가 발생했을 경우, 아래 절에 있는 방법대로 처리한다. 에러가 발생하는 조건과 처리 방법은 다음 절에서 설명한다.

- 에러 처리

- ✓ 프로그램 수행 중 다음과 같은 상황이 발생하면, `parse_url()` 함수에서 지정된 값으로 반환한다. 그러면 `main.c`에 의해 적합한 에러 메시지가 표준 출력으로 출력되게 된다.
- ✓ 처리할 예외 사항은 `parse_url.h`에 정의된 다음 세 종류가 있다.

```
#define ERR_NO_FILE      (-1)
#define ERR_NOT_ALLOWED (-2)
#define ERR_INVALID_URL (-3)
```

- ◆ input\_url로 지정한 파일이 존재하지 않았을 때
    - ERR\_NO\_FILE
  - ◆ input\_url로 지정한 파일이 존재하나, 이용할 권한이 없을 때
    - ERR\_NOT\_ALLOWED
  - ◆ 잘못된 파일 프로토콜을 사용했을 때
    - ERR\_INVALID\_URL
  - ◆ 예를 들어 전달받은 인자가 잘못된 파일 프로토콜을 사용한 경우 parse\_url() 함수의 반환값으로 ERR\_INVALID\_URL을 이용하면 된다.
- ✓ 혹시 예외 상황이 발생해서 프로그램을 마치게 되면 동적으로 할당한 메모리를 직접 해제하고 open()한 파일을 close()한다.

#### 4. Background and Additional Information

- 리눅스는 Man-page (manual-page)를 통해 상세한 메뉴얼을 제공한다.
  - ✓ 셸에 "\$ man 2 open" 를 입력하면, open() 시스템 콜에 관한 메뉴얼이 제공되며, open() 함수를 이용하는데 필요한 헤더 파일, 함수 인자, 함수 반환형 등의 정보가 제공된다.
  - ✓ 어떤 특정한 시스템 콜에 대한 메뉴얼을 보고 싶으면 "\$ man 2 syscall\_name" 을 이용하고, 평소엔 "\$ man something"을 입력하면 메뉴얼을 빠르게 참조할 수 있다.
- printf()를 이용한 디버깅은 매우 쉬우며 효과적인 방법의 하나이다.
  - ✓ 의도대로 동작하지 않는 부분은 printf()를 이용한 디버깅을 적극 권장한다.
  - ✓ 하지만, 최종 제출본 안에 해당 함수를 사용하지 않아야 한다.
- 표준 출력에 문자를 쓰기 위해 write() 함수를 이용할 수 있다.
  - ✓ write(1, buffer, sizeof(buffer)); 를 수행하면 표준 출력에 buffer에 담긴 내용을 내보낸다. 파일 디스크립터(file descriptor) 0, 1, 2는 차례대로 표준 입력/표준 출력/표준 에러를 의미한다.

- 필요한 library 함수는 직접 구현할 수 있다.
- 잘 모르는 사항은 TA에게 질문하거나, 구글을 이용한 검색하기를 권장한다.
- 표준 입력/표준 출력/표준 에러는 자동으로 open()된 상태로 프로그램이 시작하고, close()를 이용해 닫지 않아도 된다.

## 5. Restriction

- 과제는 본인이 직접 설치한 리눅스 환경에서 구현한다.
  - ✓ 테스트 서버에서 컴파일, 동작에 실패한 코드는 과제 제출물로 인정하지 않는다.
  - ✓ 테스트 서버는 Linux kernel 3.0-32bit, gcc 4.6.1 를 이용한다.
- 구현을 완료하면 <http://sys.skku.edu>에 과제를 제출하고, 수행하여 결과를 확인한다.
- 과제의 평가는 컴파일/실행 가능 여부 및 출력 결과, 작성한 문서에 의하여 평가된다.
- 시스템 콜을 사용하여 프로그램을 작성하며, Standard C library의 malloc(), free(), getenv() 를 제외한 library는 사용하지 않는다.
- 파일 입출력의 경우 open(), read(), write(), close(), lseek(), stat(), fstat() 시스템 콜 중 필요한 것을 선택하여 사용한다.
- 필요한 경우 함수를 직접 parse\_url.c 안에 구현하여 추가할 수 있다.
- parse\_url.c 이외의 파일은 수정할 수 없고, 설사 수정해서 제출하더라도 채점 서버에서 자동으로 제외하여 컴파일한다.

## 6. Skeleton Codes

- 본 과제 수행을 위하여 아래와 같이 5개의 파일이 주어진다.
  - ✓ Makefile                    GNU make utility를 위해 사용되는 파일.
  - ✓ main.c                        구현한 함수를 호출하고 에러 발생시 문자열을 출력하는 main()을 포함한 파일
  - ✓ parse\_url.h                 위에서 정의한 에러 규격과 함수를 선언한 파일
  - ✓ parse\_url.c                제시된 함수를 구현해 넣을 파일

## 7. Hand in instruction

- 작성한 프로그램 코드 상단에 이름과 학번을 주석으로 표기한다.
- 작성한 과제 코드는 zip 혹은 tar.gz 형태로 압축해 <http://sys.skku.edu>에 제출한다.

- ✓ 작업 디렉토리에서 "\$ make tar" 명령을 이용해 파일을 tar.gz으로 압축할 수 있다.
- 작성한 과제 코드와 별도로 구현에 관한 내용을 담은 보고서를 별도로 제출한다.
  - ✓ 보고서에는 프로그램의 구조를 나타낸 순서도 또는 그림을 넣는다
  - ✓ 보고서 형식은 PDF 포맷으로 제한하고, 역시 <http://sys.skku.edu>에 제출한다.

## 8. Logistics

- 과제 제출 결과는 <http://csl.skku.edu/SSE2030F13/Assignments> 에서 확인할 수 있다.
- 과제 제출 기한은 <http://sys.skku.edu> 서버시간을 기준으로 하며, 기한 이후 24시간 내에 제출할 경우 30%, 48시간 내에 제출할 경우 60%감점한다. 그 이후 제출이 불가능하며, 0점 처리한다.
- 과제에 대한 의논은 함께 할 수 있으나, 프로그램 소스코드 작성은 스스로 해야 한다.
  - ✓ 다른 사람의 과제를 copy 한 경우, 두 사람 모두 0점 처리하며 **학점상의 불이익이 있다.** 인터넷 등에서 찾은 소스코드를 그대로 copy 한 경우에도 0점처리한다. **두 번 이상 적발되면 F 학점을 받는다.**