

## Programming Assignment #4:

File transfer over network

Due: 1st Dec. (Mon), 11:59 PM

### 1. Introduction

이번 과제에선 파일을 전송하는 서버와 클라이언트를 제작한다. 이 과제의 주요 목표는 소켓 프로그래밍을 배우는 것이다.

### 2. Problem specification

클라이언트의 요청에 따라 파일을 전송 하는 서버를 서로 다른 두 개의 실행 파일로 제작한다.

#### 2.1. Client

클라이언트는 argument 없이 실행된다. 클라이언트는 표준 입력으로부터 문자열을 읽은 뒤, 해당 문자열을 분석하여 해당 서버로 접속하여 파일을 받는다.

사용자의 입력은 다음 형태를 따른다: Host[:port]/pathname

- Host의 경우, IP 주소(dotted decimal notation, 예를 들어, 1.2.3.4), domain name(예를 들어, swex2.skku.edu), 또는 localhost(127.0.0.1)가 가능하다.
  - (참고) localhost는 컴퓨터에서 "자기 자신" 을 의미하는 주소이다.
- Port는 접속할 서버의 포트 번호이고, 생략 가능하다. 생략된 경우, 기본 포트 번호 10001을 사용한다.
- Pathname은 경로로서, 전송 받을 파일의 서버에서의 (상대) 주소를 표현한다.
- 한 입력은 200바이트를 넘지 않는다.
- 각 입력은 개행 문자(\n) 을 기준으로 구분한다.

입력 예)

```
1.2.3.4:10001/data.txt
localhost/data.txt
swex2.skku.edu:50000/data.txt
```

각 입력을 분석하여, 상대방의 주소, 포트, pathname을 분석하여, 서버로 접속한다.

예를 들어, swex2.skku.edu:50000/dir/file 이란 입력의 경우 다음과 같이 분석할 수 있다:

```
Host: swex2.skku.edu
Port: 50000
Pathname: dir/file
```

유효하지 않은 입력이 들어온 경우, 해당 입력을 무시한다. 유효하지 않은 입력이란, 다음 상자의 규칙을 만족하지 않는 입력이다.

```
string1 : num / string2
```

여기서 `string1`은 서버의 주소, `num`는 포트 번호, `string2`는 `pathname`이 될 것이다.

(프로토콜은 생략될 수 있기 때문에 콜론과 `num`은 생략 가능)

예외로, 표준 입력으로부터 `exit` 란 문자열이 들어오면, 클라이언트 프로그램을 정상 종료한다.

서버와 통신 방식은 2.3절의 프로토콜을 따른다. 만약 에러가 발생하면, 그에 따른 적절한 함수를 호출한다. 서버로부터 파일을 전송 받을 경우, `pathname`에서 디렉토리를 제외한 파일 이름에 해당하는 새로운 파일을 열어 받은 내용을 저장한다.

- 예를 들어 `pathname`이 `dir/file` 일 때, 서버로부터 해당 파일을 다운로드 받을 경우, 현재 디렉토리에 `file` 이란 파일을 생성하여 저장한다.
- 파일이 존재하면 덮어쓴다. (`O_TRUNC` 플래그)

매번 사용자의 입력에 따라 서버로 요청을 보낸 뒤, 결과 메시지를 표준 출력으로 출력한다. 결과 메시지는 아래 다섯 종류가 있다.

서버로부터 정상적으로 파일을 받았음	OK, <pathname>
서버로부터 에러 코드를 받았음 (ENOENT)	E_FILE, <pathname>
서버로부터 에러 코드를 받았음 (EPERM)	E_PERM, <pathname>
서버로 연결할 수 없음	E_CONN, <pathname>

만약, `pathname`이 `dir/file` 이었을 때, 해당 파일의 다운로드를 서버로 요청했지만 해당 서버로 연결할 수 없었다면 다음 메시지를 표준 출력으로 내보낸 뒤 다음 입력을 받는다.

```
E_CONN, dir/file
```

## 2.2. Server

서버는 한 개의 `argument`를 받아 실행되는데, 해당 `argument`는 서버로서 동작할 포트 번호이다. 만약 `argument`가 없다면, 기본 포트로 10001을 사용한다. 서버를 생성한 다음 클라이언트와의 연결이 수립되면, 새로운 자식 프로세스를 생성한 뒤 해당 프로세스가 그 연결을 담당하도록 한다.

서버는 클라이언트로부터 파일의 상대 경로를 받은 뒤, 해당 파일을 열어 클라이언트에게 전송해준다. 구체적인 방식은 다음과 같다.

- 클라이언트가 보내는 경로 `pathname`은 개행 문자(`\n`)로 끝나는 문자열이다.
- 해당 경로에 해당하는 파일을 현재 작업 디렉토리(`CWD`)에서부터 찾아서 연다.
- 파일을 연 뒤, 해당 파일의 내용을 읽고, 클라이언트와 연결된 소켓으로 전송한다.
- 전송이 완료되면, 소켓과 파일을 닫고 해당 프로세스를 종료한다.

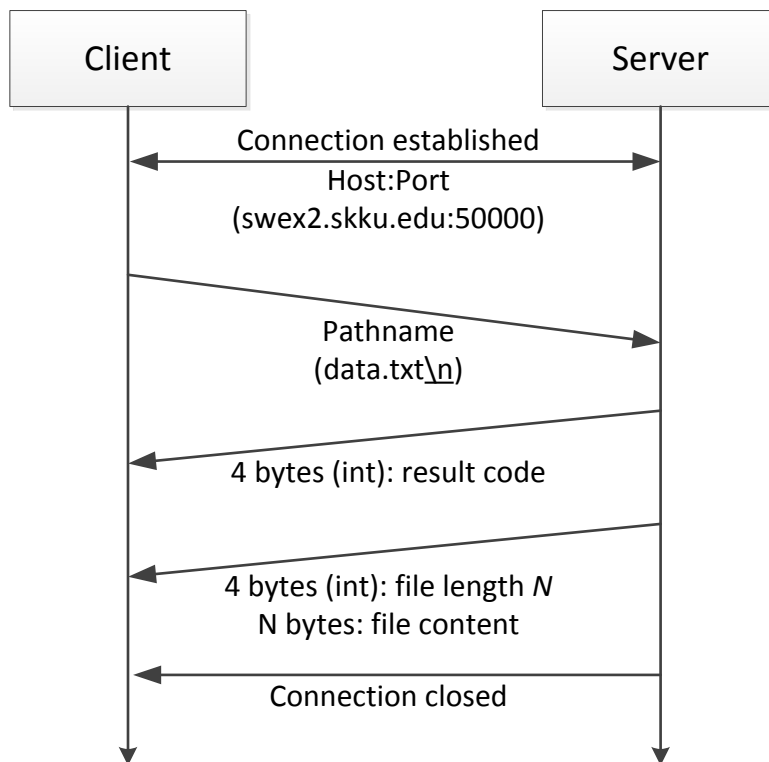
구체적인 전송 프로토콜은 2.3장에 기술한다.

서버의 자식 프로세스는 클라이언트의 요청 하나를 처리하기 위해 생성하고, 파일의 전송이 완료되면 해당 프로세스를 종료시킨다. 자식 프로세스가 종료되었다는 시그널이 연결을 기다리던 부모 프로세스로 전달되면, 부모 프로세스는 자식 프로세스를 정리한다.

서버 프로세스는 스스로 종료되지 않고, 계속 동작하도록 한다. 만약, 어떤 이유든지 지정된 포트로 서버를 생성할 수 없다면, 프로그램을 종료하며 -1 을 반환한다.

### 2.3. Protocol

다음 그림은 본 프로그램의 서버와 클라이언트가 지켜야 할 프로토콜을 표현한 것이다.



#### 2.3.1. Pathname

- 클라이언트가 전송하는 pathname은 개행 문자로 끝나는 문자열이다.
- 서버는, 해당 파일을 찾아서 연다.
  - 만약 서버의 CWD가 /home 디렉토리이고, 클라이언트로부터 받은 pathname이 data.txt 이라면, 해당 파일은 /home/data.txt 일 것이다.
- 파일의 상태에 따라 적절한 result code를 전송한다.

### 2.3.2. Result code

실행 뒤 결과는 4 바이트 숫자로 표현된다. Result code의 종류는 다음 세 종류가 있다.

- 0 ← 파일을 정상적으로 읽을 수 있음; 앞으로 파일을 전송할 것임
- 1 ← 해당 파일이 존재하지 않음 (ENOENT)
- 2 ← 해당 파일을 사용할 권한이 없음 (EPERM)

Pathname에 해당하는 파일을 정상적으로 열리면, 숫자 0을 보낸다. 만약, open 시도 시 해당 파일이 존재하지 않아서 `errno`의 값이 `ENOENT` 이라면, 숫자 1을 보낸다.

파일이 정상적으로 열렸다면, 소켓으로 4 바이트 파일 크기와 파일의 내용을 전송한다. 파일 전송을 완료하면, 소켓과 파일을 닫고 해당 자식 프로세스를 종료한다. 만약, 에러가 발생했다면 `code`를 보내는 것으로 클라이언트의 요청을 다 처리한 것이니, 바로 소켓을 닫고 프로세스를 종료한다.

네트워크로 정수형 데이터를 전송할 땐, 반드시 **network endian**으로 보낸다.

또한, TCP/IP 환경에서, 최근에 닫힌 포트의 경우 일정 시간 동안 사용 제약이 걸릴 수 있다. 해당 포트로 바인딩이 되지 않는 경우, 다른 포트 번호를 이용한다.

본 문서에서 언급되지 않은 부분, 예를 들어 사용자에게 입력을 요청 받는 것 등은 자유롭게 꾸밀 수 있다.

### 3. Restrictions

- 과제 구현을 위해 여태까지 배운 리눅스 시스템 콜/라이브러리 함수를 이용한다.
- 동적 메모리 할당 함수를 제외한 다른 라이브러리 함수는 사용할 수 없다. 필요하다면 직접 구현하여 사용한다.
  - 쉬운 디버깅을 하기 위해 라이브러리 함수를 사용할 수 있다. 다만, 제출한 파일엔 해당 함수들이 직접 사용되지 않아야 한다. (주석 등으로 처리해도 괜찮음)
- 어떤 자원을 동적으로 할당 받았다면, 프로그램 종료 전에 반드시 해제해야 한다.
  - 여기서 자원이란 파일, 메모리, 자식 프로세스, 소켓을 뜻한다.

### 4. Hand in instruction

- 작성한 코드 상단의 주석에 이름과 학번을 작성한다.
- `make` 명령을 통해 프로그램이 제대로 만들어지는지 확인한다.
  - 제출한 코드가 컴파일되지 않을 경우, 경우에 따라서 **해당 제출은 채점되지 않거나**

**심한 페널티를 받게 될 것이다.**

- 프로그램 코드를 "학번.tar.gz" 형태로 압축한다.
- 본 과제 수행 시 구현 방법과 디자인을 설명하는 보고서를 PDF 포맷으로 작성하여 "학번.pdf" 이란 이름을 붙인다. (가능하면 PDF가 가장 좋지만, 대중적인 문서 포맷은 다른 포맷도 괜찮음)
- 과제를 제출하기 위해 [[wooyeong at cs.skku.edu](mailto:wooyeong@cs.skku.edu)] 주소로 메일을 보낸다. 메일 전송 시 전송한 코드 파일과 문서 파일을 각각 첨부하고(압축하지 말 것!), 메일 제목은 다음과 같이 명명한다:

[SWE2007] PA #4, 학번, 이름

## 5. Logistics

- 본 과제는 혼자 수행한다.
- 제출 상태는 과목 홈페이지 <http://cs.skku.edu/SWE2007F14/Projects> 에 즉각적으로 공지 될 것이다.
- 과제 제출 시간은 메일 도착 시간을 기준으로 하며, 과제를 지연 제출하면 **기한 직후엔 10%가 감점되고, 매 24시간마다 10%씩 추가로 감점된다.** 감점폭이 50%를 초과하는 순간부터 더 이상 제출을 받지 않는다.
- 다른 사람의 과제를 copy할 경우, 개입한 사람 전부 해당 과제에 대해 0점 처리되고, 교수님께 보고되며, **성적 산정에 불이익이 있다.** 또한, copy가 두 차례 이상 적발될 경우 F 학점이 부여될 수 있다.

Have fun!

---

정우영, 담당 조교  
컴퓨터시스템연구실