

System Programming and Practice

Assignment 4 : Remote Development Kit

Background

원격 개발도구 - Remote Development Kit

소프트웨어 개발과 엔지니어링은 본인이 소유한 랩탑 혹은 데스크탑에서도 할 수 있지만, 실무에서는 거대한 시스템을 개발해야하는 경우가 있다. 이러한 시스템을 빌드하기 위해서는 머신의 퍼포먼스가 충분히 받쳐줘야하는 경우도 있고, 혹은 개발 환경이 리눅스로 제한되는 경우도 더러 있다. 때문에 적지않은 개발자가 SSH 접속을 통해 원격 기기의 시스템에 접속하여 Vim을 통한 개발을 해 왔다.

Vim은 매우 생산성이 좋은 에디터이며, 여러가지 플러그인을 통해 멋진 개발도구로 만들 수 있지만, 최신 IDE에 비해 진입벽이 높다는 단점이 있어 최근 에디터 중에서는 SSH를 통한 원격개발을 지원하고 있다. 일례로, VSCode는 Remote Development Extension을 제공하고 있으며, IntelliJ는 SSH Development 기능을 지원하고 있다.

Goal

본 과목을 수강하는 학생들도 라즈베리파이 실습에 VSCode의 원격개발도구를 활용하는 학생이 많다. 그러나 원격개발도구의 백엔드 프로세스는 임베디드 프로세서에서는 다소 무거운 프로세스일 수 있다. 따라서 본 과제에서는 컴파일할 코드만을 전송하고 실시간으로 컴파일하여 매우 가벼운 원격개발을 지원하는 원격개발도구를 직접 만들어본다.

Overview

구현을 구체적으로 생각해보기 위해 다음 시나리오를 참조하자.

1. 자신의 컴퓨터에서 라즈베리파이에서 실행할 코드를 작성한다.
2. 클라이언트 프로그램에 해당 코드파일과 접속할 라즈베리파이의 IP주소와 포트번호를 실행 아규먼트로 입력한다.
3. 클라이언트 프로그램이 해당 라즈베리파이의 서버 프로그램과 소켓 연결을 시도 후, 코드 파일을 전송된다.
4. 서버 프로그램은 코드 파일을 받아 gcc 명령을 통해 컴파일 후 실행 파일을 실행한다.
5. 이후 클라이언트 프로그램의 `stdio`, `stdout`은 마치 자신이 작성한 코드가 실행되는 프로그램의 `stdio`, `stdout`을 이용하는 것처럼 보여야한다.
 - a. 서버프로그램의 `stdin`과 `stdout`을 클라이언트 소켓 스트림에 `redirect` 한다.
 - b. 클라이언트 프로그램의 입력을 라즈베리파이의 서버 프로그램에 전송하고, 서버 프로그램으로부터 전달받은 내용을 출력한다.

Requirement

클라이언트 프로그램

- i. 접속할 라즈베리파이의 IP주소, 포트번호, 파일의 경로를 아규먼트로 받는다.
- ii. 코드 파일을 서버에 전송한다.
- iii. 서버 측에서 전달된 코드의 실행 내용을 출력한다.
- iv. 입력을 서버 프로그램으로 전달한다.
- v. 서버로부터 EOF를 수신하면 프로그램을 종료한다.

서버 프로그램

- i. 서버 프로그램은 백엔드로써 작동해야하기 때문에, 한 클라이언트와 연결 수립/해제 이후에도 다시 클라이언트와 연결할 수 있도록 한다.
- ii. 클라이언트와 연결 이후 전달 받은 코드파일을 임의의 경로(혹은 서버프로그램이 존재하는 경로)에 저장하고, 해당 파일로 system 함수를 통해 gcc 컴파일 명령을 수행하도록 한다.
- iii. 클라이언트 프로그램과 연결된 순간부터 stdout으로 출력되는 메시지를 전부 클라이언트 측 소켓으로 redirect한다.
- iv. 클라이언트 프로그램으로 부터 전달 받는 모든 입력을 stdin으로 redirect한다.
- v. 클라이언트 연결 해제 이후 시스템 메시지의 출력이 정상적으로 서버에서 출력되도록 redirect된 file descriptor를 다시 원래대로 돌려놓는다.
- vi. 서버에서 실행한 프로그램이 종료되면 클라이언트에 EOF를 송신 후 다시 연결 대기 상태로 돌아간다.

Submission

과제 제출은 다음과 같이 제출하도록 한다.

1. 프로그램과 코드를 각각 program, code 폴더로 구성한다.
2. 두 폴더를 자신의 학번을 이름으로 하여 압축하여 제출한다.
3. 본 과제의 경우, 보고서를 별도로 작성하지 않기 때문에 코드에 주석을 충분히 작성하도록 한다.

압축파일의 구성 예시

- 압축파일 - 학번.zip (e.g. 202224210.zip)
 - program
 - client
 - server
 - code
 - client.c
 - server.c

Grading Criteria

본 과제는 총 10점으로, 다음 기준을 토대로 채점이 진행된다.

1. 클라이언트와 서버간 통신을 구현 (1)
2. 클라이언트가 서버로 코드파일을 전송하여, 서버에서는 해당 코드파일을 받아 파일 형태로 저장할 수 있음 (1)
3. 서버에서 gcc 컴파일 명령을 수행할 수 있음 (1)
4. 서버 프로그램에서 redirect를 과제 요구사항에 지시된 내용대로 활용하였음 (4)
5. 스레드를 적절히 사용하였음 (2)
6. 클라이언트 및 서버 프로그램이 과제 요구사항의 흐름을 만족함 (1)
 - 주석 없이 코드만 제출할 경우, 예외없이 0점

Example

두 숫자를 인풋으로 받아 덧셈한 내용을 출력하는 간단한 프로그램을 원격개발도구로 개발하는 예시

(## 로 표시된 라인은 주석)

- 기존 프로그램의 실행 예시

```
## 프로그램 실행
$ ./code

## 숫자 입력
1 2
result: 3 ## 결과

## 프로그램 종료 후 콘솔로 복귀한 상태
$ _
```

- 위 프로그램의 코드를 본 과제의 원격개발 프로그램에 활용한 예시

Client

```
$ ./client 192.168.0.2 24210 code.c

## 서버와 연결 이후 콘솔에 입력
1 2

## 실행된 내용이 redirect되어 전달된다
[remote] result: 3

## 수행한 프로그램이 끝나고 EOF 수신
[*] session closed

## 다시 콘솔로 복귀
$ _
```

Server

```
$ ./server

## 클라이언트 연결 대기
[*] wait for client ...
[*] client connected

## 현재 건네받은 코드가 실행중
## 원래 출력되어야 할 내용은 Client 측으로 전송되게 된다.

## 수행한 프로그램이 끝난 경우
[*] session closed

## 다시 대기
[*] wait for client ...
```

Appendix (Hint)

1. System()

system() 함수는 문자열 입력받아 Shell에서 해당 문자열을 직접 수행하는 것과 같은 명령을 할 수 있다. 내부적으로는 fork(), exec() 함수 등으로 구성되어 있으며, system() 함수를 호출한 프로세스와 동일한 표준 스트림(stdin, stdout, stderr)을 사용한다.

```
#include <stdlib.h>
int system(const char *string);
```

- reference : <https://www.ibm.com/docs/ko/i/7.3?topic=functions-system-execute-command>

2. Redirection

리다이렉션(redirection)은 컴퓨팅에서 표준 스트림을 사용자 지정 위치로 우회할 수 있는 다양한 유닉스 셸을 포함한 대부분의 명령어 인터프리터에 일반적인 명령이다. **방향 지정, 방향 변경**으로도 부른다.

간단히 말하자면, redirection은 stdin, stdout과 같은 표준 스트림을 다른 방향으로 출력하는 것을 말한다. 다른 방향을 지정하는 방법은 여러가지가 있으며, 실습에서 배운 내용을 토대로 해결할 수 있다.

- reference: <https://ko.wikipedia.org/wiki/리다이렉션>