설계 계획서

20141210 정진우, 20141375 최효석 A조 1팀

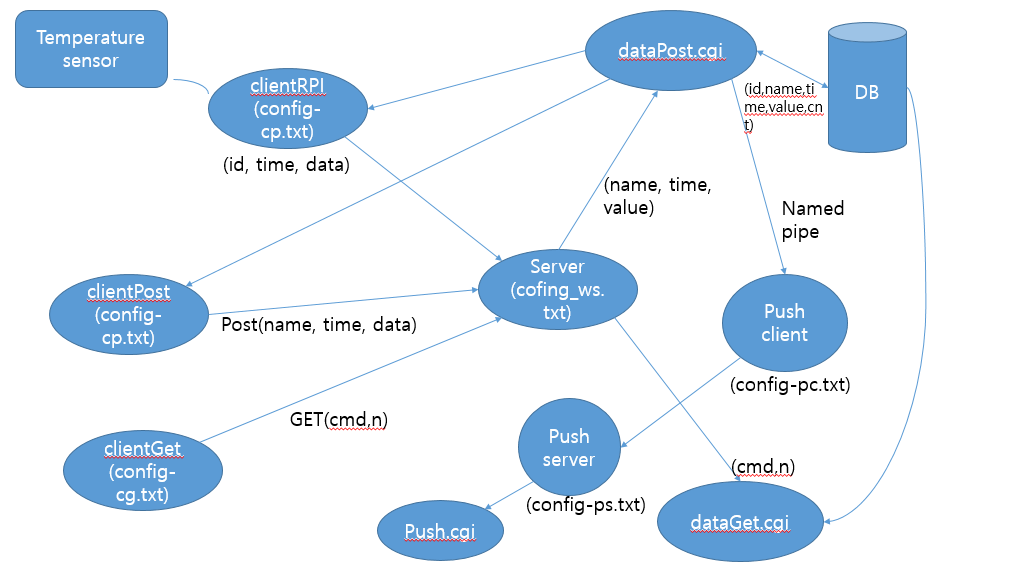
1. 문제 개요

구현할 내용과 목표

라즈베리파이를 활용한 온습도 측정 및 재난 방지 시스템 구현

1. 문제 분석

2-1 설계 구성요소



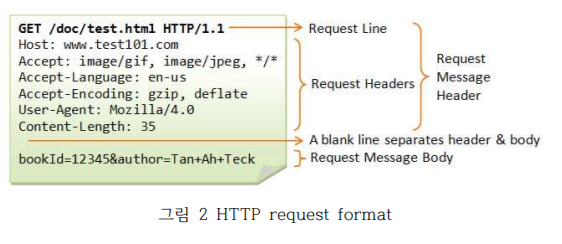
• 제공되는 소스코드 분석 (함수들 과의 관계를 중심으로)

전체적 흐름

1. ./Server를 실행시킨다.
2. ./clientGet, ./clientPost를 이용하여 서버에 http 형식의 메시지를 보낸다.

각각 config-cg.txt , config-cp.txt 를 읽는다.

1. http 형식의 메시지를 request.c 에서 처리를 한다
2. Server.c 의 consumer() 함수가 동작하는데 이 함수에서 requsetHandle()함수 진행한다
3. requsetHandle() 에서 Rio\_readlineb를 사용하여 method , uri , version 을 파일에서 정보를 먼저 읽어온다.



1. 읽어온 uri를 바탕으로 paruseURI 를 실행한다.
2. 여기서 cgi를 기준으로 정적인지 동적인지를 구분한다.
3. 정적일 경우 requestServerStatic 이실행되면서 /index.html 이 실행된다.
4. 동적 일 경우 requestServerDynamic 이 실행되며 기본적으로는 Current version does not support CGI program. 이 출력된다.

dataPost.c와 dataGet.c

10 .dataGet과 dataPost는 소스코드를 바탕으로 cgi 파일을 생성한다.

11.clientGet.c는 uri에 정보를 담아서 서버에 작업을 요청하고

12 .clientPost.c는 requestbody에 정보를 담아서 서버에 작업을 요청한다.

• 소스코드의 어떤 부분이 중요한가, 어떤 기능을 수행하는가,

Request.c 이 부분에서 동적,정적 요청인지를 판단하고 정적일 경우 해당 html을 전달하고 동적일 경우 해당하는 cgi프로그램을 실행시켜 그 결과값을 client에게 다시 전달하는 기능을 담당한다.

Sever.c : 클라이언트에게 요청을 받으면 Request.c에서 처리한다. 정적일 경우 바로 client에게 동적일 경우 cgi파일 출력을 fd를 사용해 client에게 출력

• 어떤 모듈들을 구성할 것 인가?

clientRPI,7

dataPost.cgi

DB

phshclient

pushserver

push.cgi

dataget.cgi

• 어떤 자료구조가 필요한가?

온도와 습도, 시간, 장소를 담는 구조체

센서로 읽은 정보를 담는 DB 테이블

• 어떤 알고리즘을 사용할 것인가,

1. 센서로 기기 주변의 온습도의 정보를 체크하고 전송
2. 서버에 저장하고 클라이언트로 상황을 보고
3. 같은 장소, 연속된 시간 , 특정한 온습도 이상 이 1시간 이상 지속될 경우 재난 message발생 (PUSH 알람으로 구현)
4. 반복.

• 어떤 시스템 콜 / 함수를 사용할 것인가?

동적 요청일 경우 ?뒤의 argument와 앞의.cgi 파일의 이름을 구별하기위한 **strtok** 함수

**Due2()** 함수를 이용하여 cgi 파일의 실행결과를 서버가 아닌 클라이언트로 redirect 해준다.

**Fork(), execve()**를 사용해서 sever의 CGI 프로그램 수행을 요청한 client 에게 그 결과를 전달하게 한다.

**Setenv()** 함수를 사용해서 CGI 프로그램의 argument 를 환경변수에 문자열을 할당해서 sever 가 cgi 프로그램에 argument를 정상적으로 전달될 수 있게 만들어준다.

Shell 기능의 내장 명령어를 구현해야 하기 때문에 명령어와 옵션을 구별한 strtok()함수가 필요하다.

Shell 의 Random 명령어에서 임의의 값을 요청해야 하기 때문에 **rand()**함수를 사용한다.

dataPost.cgi 가 push 알람을 사용하려면 경고를 보낼 때 named pipe를 통해 보내야하기 때문에 이를 위해

Named pipe의 생성: **mkfifo()**

생성된 named pipe의 open: **open()**

Named pipe를 통해 값을 보내고 받기: **write(),read()**

Named pipe를 더 이상 사용하지 않을 때: **unlink()**

이와 같은 각각의 함수를 사용한다.

쓰레드 풀을 만들기 위해 **ptread\_create()**를 사용하고 **accept()**를 사용해서 쓰레드와 소켓을 연결해준다.

2-2 설계 제한 요소

• 수행 환경 – 모든 프로세스들은 Linux 환경에서 표준 C 라이브러리 와 POSIX 표준 시스템 콜로 구현되어야 함. – server 프로세스는 DB가 가동될 수 있는 시스템에서 수행되어야 함 [1]. – clientPost 프로세스는 Raspberry PI에서 수행되어야 함

3.진행 일정

– 매 주차마다 어떤 내용을 설계하고, 어떤 내용을 구현하고, 어떤 내용을 테스트/디버그/검증할 것인지 기술

(3주차) 현재 주차 동적 ,정적 요청 구분 및 cgi 파일 실행

(4주차)다음주,추석 (1단계 구현완료)

1. 동적, 정적 요청 여부 구분
2. 파일명 분리
3. Cgi 프로그램 실행
4. 파일 기술자 변경을 통해 cgi 실행결과 반환

(5주차)(2단계일부 구현)

1. Web server의 소스코드 변경해 client로부터 받은 argument를 cgi프로그램에 전달
2. Cgi 프로세스가 argument를 web server로부터 받도록 수정

(6주차)1단계 발표 ,(2단계 구현 완료 ,3단계 일부 구현)

1. Command shell 형태로 clientPost가 사용자로부터 입력을 받고 web server로 request를 보내는 기능 추가.(help, name , value)

(7주차) (3단계 일부 구현완료 및 디버깅)

1. Command shell 형태로 clientPost가 사용자로부터 입력을 받고 web server로 request를 보내는 기능 추가.(send , random, quit)

(8주차)중간고사(4단계 일부 구현 )

1. Server컴퓨터에 DB설치
2. 필요한 table 구성

(9주차) (4단계 구현 완료 및 5단계 일부 구현)

1. dataPost.cgi 가 환경변수를 통해 전달받은 argument 들의 정보를 DB에 저장 하도록 코딩(4단계 완료)
2. clientGet 실행하면 prompt를 이용해 몇가지의 명령어를 수행할 수 있도록 확장( LIST, INFO ,GET , quit, exit)

(10주차)4단계 발표(5단계 구현 완료, 6단계 일부 구현)

1. dataGet.cgi 프로그램이 argument를 바탕으로 DB에서 정보를 읽어 client 에 회신(5단계)
2. push Server구현(6단계)

(11주차)5단계 발표(6단계 구현 완료)

1. push Client 구현
2. multi-thread구현

(12주차) (7단계 구현 완료)

1. clientPost -> clientRPI.c 라즈베리파이 변경

(13주차)(8단계 구현 완료)

1. web server를 thread pool로 변경

(14주차) 마지막 8단계 발표

1. 기능추가
2. 참고 문헌

CGI: https://terms.naver.com/entry.nhn?docId=1179717&cid=40942&categoryId=32854