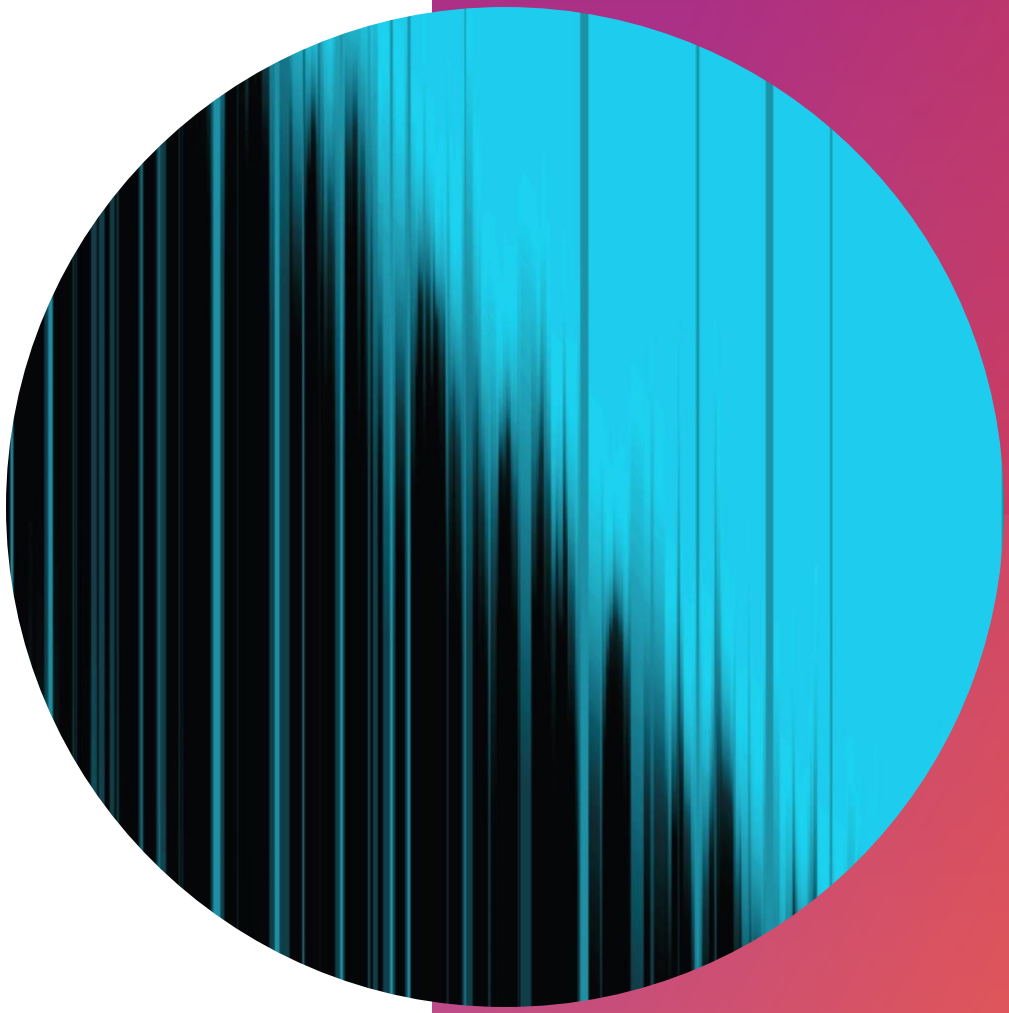
HTTP Daemon



ONEM2M TINY IOT PROJECT

엄경호

지난 진행 상황

I. Interop 후 보완 사항 적용

응답 헤더

Interop 때 다양한 응답 헤더 누락 (RSC, RVI, RI 등등..)

httpd.h 헤더의 DEFAULT_RESPONSE_HEADERS를 수정하면 응답 헤더에 반영되도록 구현

```
#define RESPONSE_PROTOCOL "HTTP/1.1"
#define DEFAULT_RESPONSE_HEADERS "Connection: Close\nAccept: application/jso
```

응답 헤더 세팅 함수 구현

```
void set_response_header(char *key, char *value) {
  char header[1024];

  sprintf(header, "%s: %s\n", key, value);
  strcat(response_headers, header);

  return;
}
```

```
char *header_value;
if((header_value = request_header("X-M2M-RI"))) {
    set_response_header("X-M2M-RI", header_value);
}
if(header_value = request_header("X-M2M-RVI")) {
    set_response_header("X-M2M-RVI", header_value);
}
```

응답 헤더

Content-Length 헤더 추가를 위해 http 응답 함수 구현

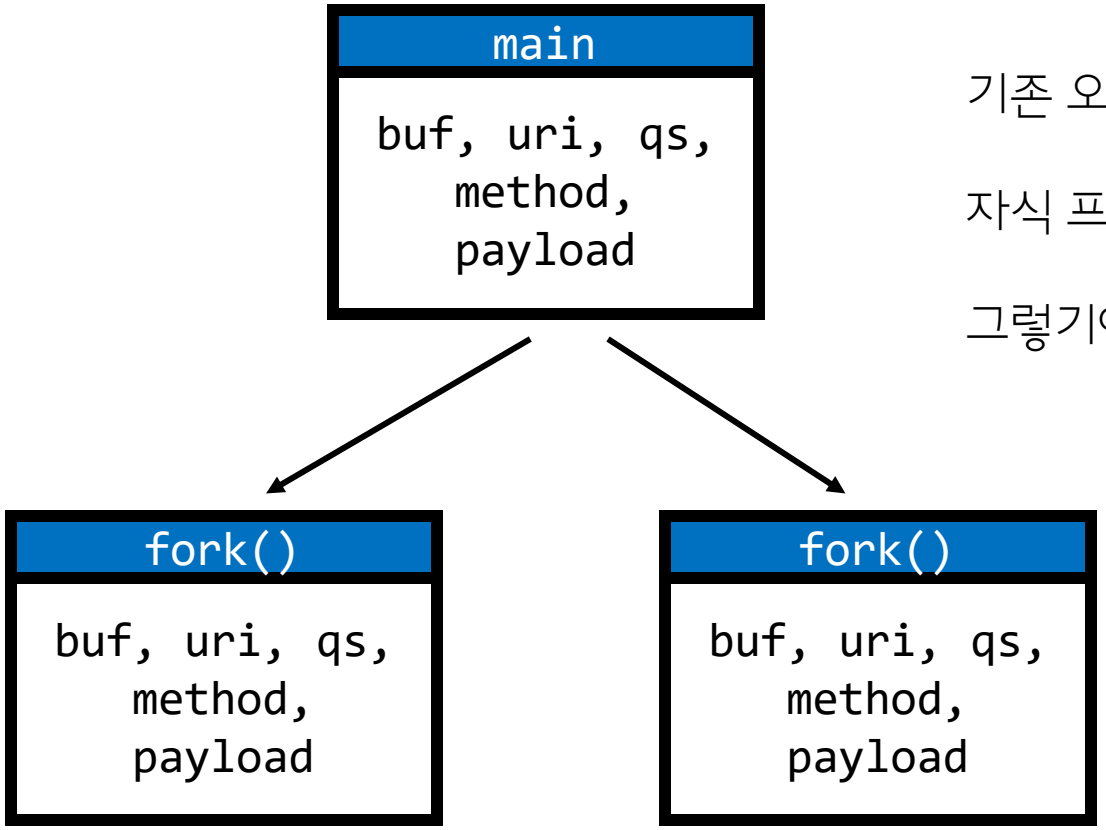
```c
void respond_to_client(int status, char *json) {
  if(json) {
    if(response_json) free(response_json);
    response_json = (char *)malloc((strlen(json) + 1) * sizeof(char));
    strcpy(response_json, json);
  }

  if(!response_json) {
    fprintf(stderr,"response_json is NULL\n");
    return;
  }

  char content_length[16];

  sprintf(content_length, "%ld", strlen(response_json));
  set_response_header("Content-Length", content_length);

  switch(status) {
    case 200: HTTP_200; break;
    case 201: HTTP_201; break;
    case 209: HTTP_209; break;
    case 400: HTTP_400; break;
    case 403: HTTP_403; break;
    case 404: HTTP_404; break;
    case 406: HTTP_406; break;
    case 413: HTTP_413; break;
    case 500: HTTP_500; break;
  }
  printf("%s",response_json);
  //free(response_json);
}
```

```c
response_json = acp_to_json(gacp);
HTTP_200;
printf("%s",response_json);
```

```c
response_json = acp_to_json(gacp);
respond_to_client(200, NULL);
```

```c
respond_to_client(200, "{\"m2m:dbg\": \"resource is deleted successfully\"}");
```

Multi thread 관련

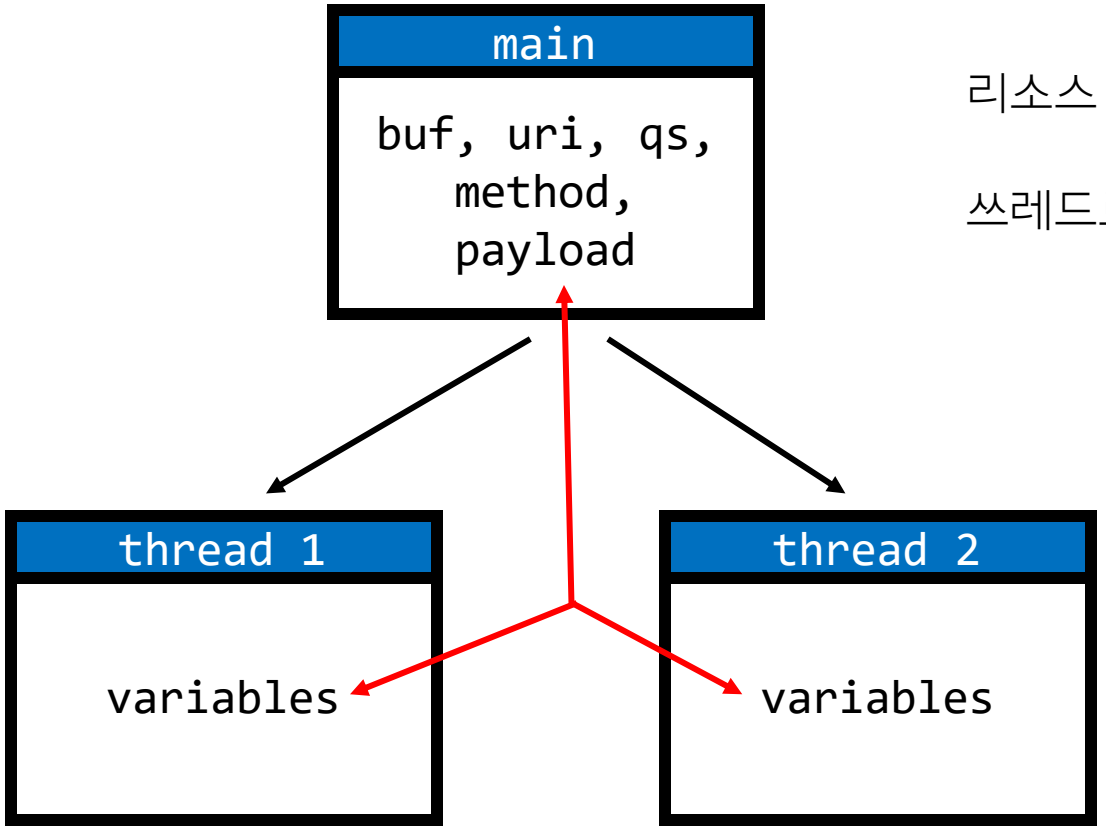


기존 오픈 소스에서는 fork 함수를 사용하여

자식 프로세스에서 응답을 처리하였음

그렇기에 변수는 1개만 선언되어 있어도 독립적으로 접근 가능

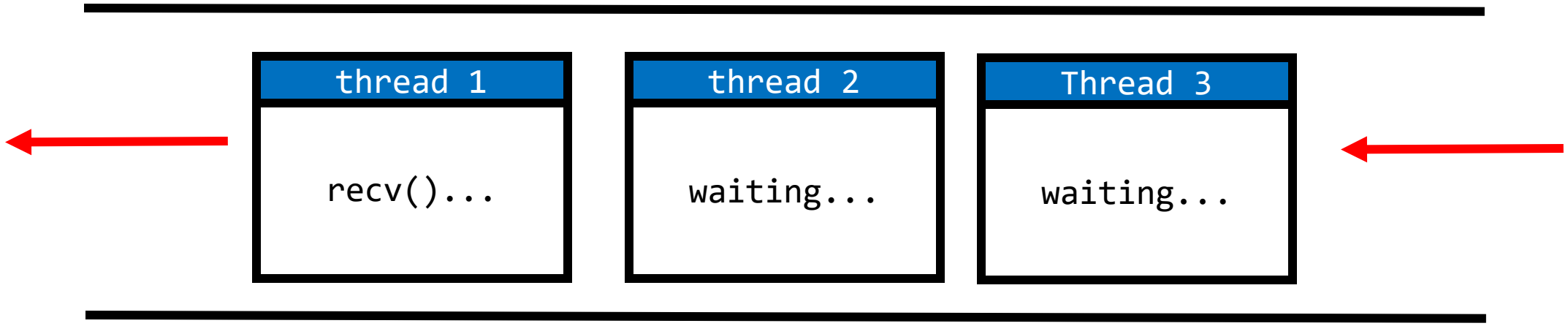
Multi thread 관련



리소스 트리 변경 사항이 반영되지 않는 오류로 인해

쓰레드로 변경했었음

Multi thread 관련



Mono thread로 동작하게 되면서 recv() 함수에서 병목이 생기면 서버가 다른 응답을 처리 못했음

Multi thread 관련

Mono thread

buf → accept(), recv()

method, uri, qs, request header parsing

oneM2M Operation

기존에는 모든 변수를 1개씩 두고 mono thread로 동작하였기에 동시 접근 문제가 없었음

Multi thread 관련

buf[MAX_CONNECTION]

Mutex Lock

method, uri, qs, request header parsing

oneM2M Operation

Mutex Unlock

recv() 병목 제거를 위해 우선 buf만 쓰레드 별로 할당할 수 있도록 배열로 수정

config 파일 관련

And just a small comment, it would be good to have a config file where to find all CSE related parameters (I think it does not exist yet, at least, I have not seen it). I know those parameters as we were running the interop, otherwise I wouldn't have known.

CSE 생성 후에 해당 정보를 가독성있게 config 파일로 저장하는 형태 ?

또는

config 파일을 기반으로 CSE 를 생성하는 형태 ?
그렇다면 기존에 생성되어 있는 CSE는?

속성 값이 NULL인 경우

NULL인 속성이 JSON 속성 값으로 포함 되는 오류가 있었지만 원인은 이미 다 알고 있는 상태이기 때문에 수정할 방향만 정해진다면 바로 수정이 가능한 상황