진예지
최재훈

Table of Contents

#1 프로젝트 팀 구성

1334609 최재훈

팀장

Web <-> Server 동작 및 Raspberry pi 개발

1334333 진예지

팀원

Web Interface 및 Design

#2 Gantt Chart

	9월	10월					11월				
	9.3 ~	1주	2주	3주	4주	~31	1주	2주	3주	4주	~29
사전 조사											
요구 분석											
기능 설계											
WEB개발											
RaspberryPi 개발											
SERVER 개발											
테스트											

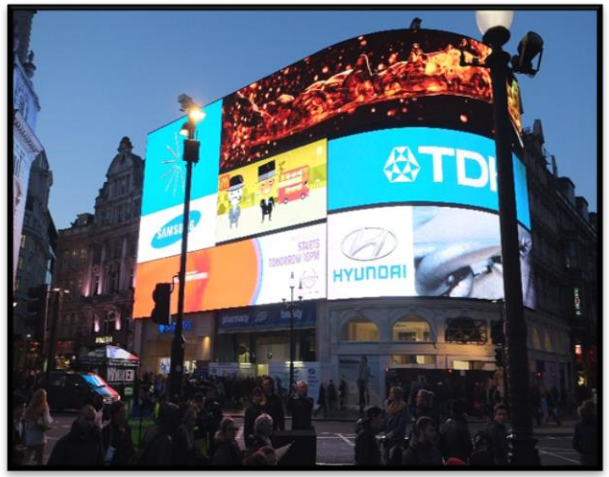
#3 개발 동기

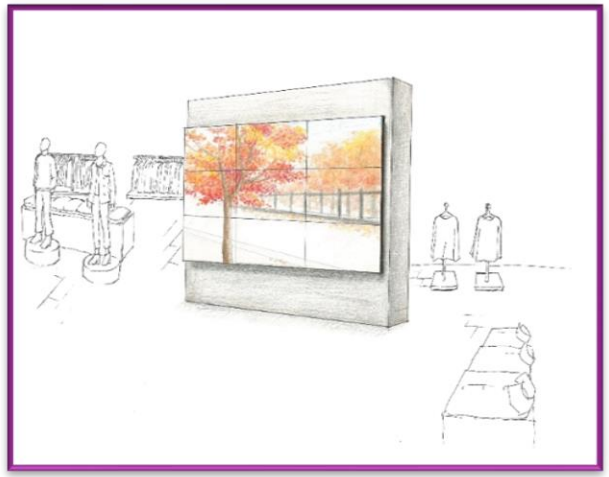
Digital Signage?

✓ 디지털 디스플레이(LCD, LED)를 공공장소나 상업공간에 설치하여 정보, 엔터테인먼트, 광고 등을 제공하는 디지털 미디어

✓ 소형 액정을 사용한 상점의 POP 광고

~ LED를 활용한 대형 옥외비전 광고





#3 개발 동기

현 시스템의 문제

- ✓ USB를 사용하여 수작업으로 영상을 교체 하기때문에 사용에 번거로움이 있음.
- ✓ 동작할 수 있는 기능에 비해 무겁고 공간을 많이 차지하는 스크린과 기기
- ✓ 프렌차이즈를 제외한 자영업자는 Digital Signage 서비스 이용이 힘듦.

➡ 높은 서비스 이용 비용

#4 개발 목표

1. 대용량 저장장치 및 웹 호스팅 기능을 가진 NAS를 이용.

2. 사용자가 Raspberry Pi 의 관리를 용이하도록 설계.

3. 사용자가 접속한 WEB을 통한 실시간 Raspberry Pi 제어.

#5 개발 환경

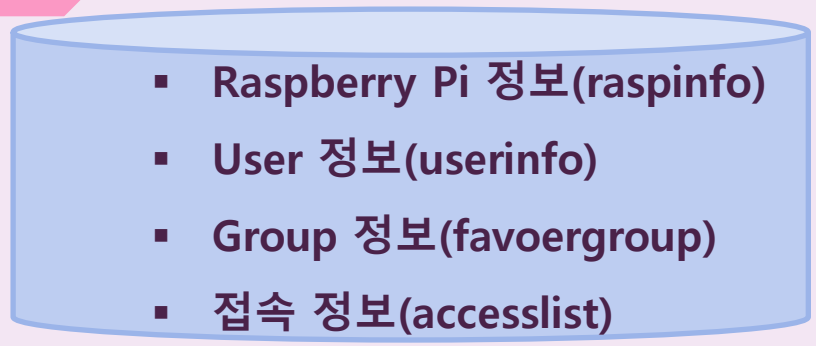
언어

Python3 – Raspberry Pi
JAVA – SERVER
PHP, CSS, JavaScript, Jquery – WEB
MySQL (Maria DB) - DB

API

Google – Google Maps API

DB



#6 요구 사항 분석

1. WEB을 통한 다수의 Raspberry Pi를 컨트롤

- ✓ IP를 이용하여 소켓통신

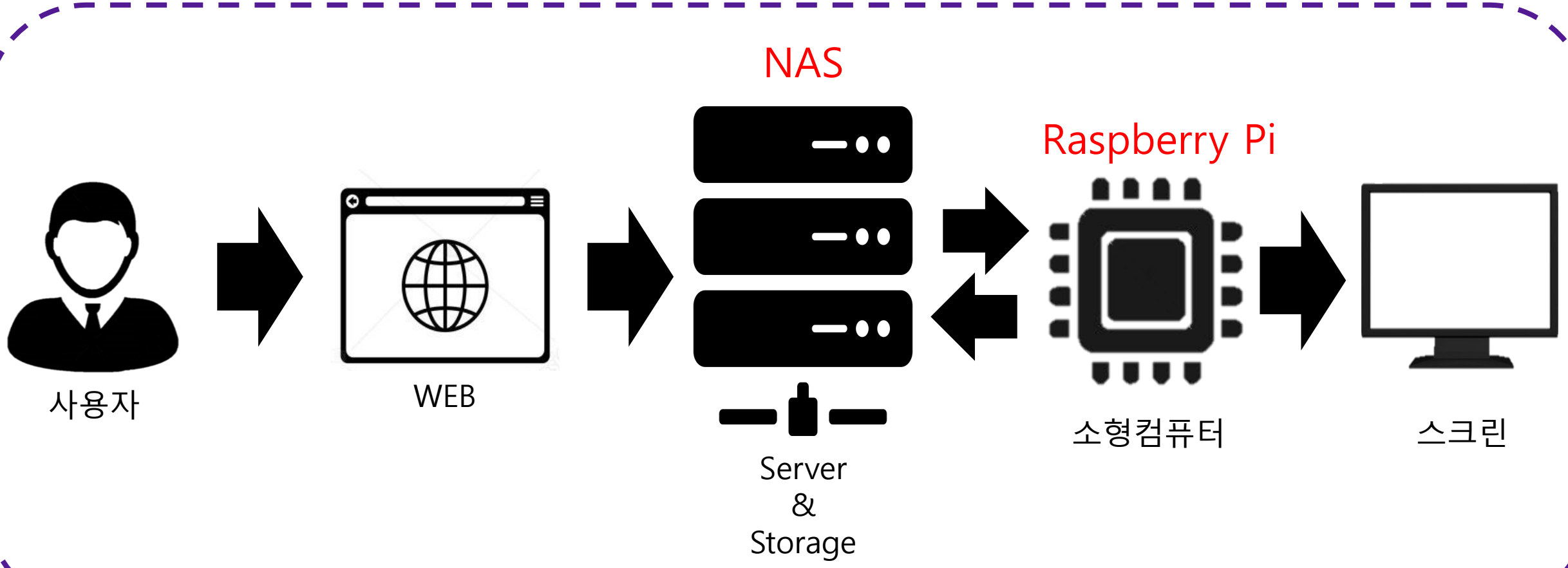
2. NAS 서버 구축 및 웹 호스팅

- ✓ PHPMyAdmin을 이용
- ✓ MySQL과 호환 되는 Maria DB 이용

3. Raspberry Pi 위치 파악

- ✓ Google Maps API를 이용하여 해당 RaspBerry Pi 위치 표시.

#7 스토리보드 및 시스템 구성

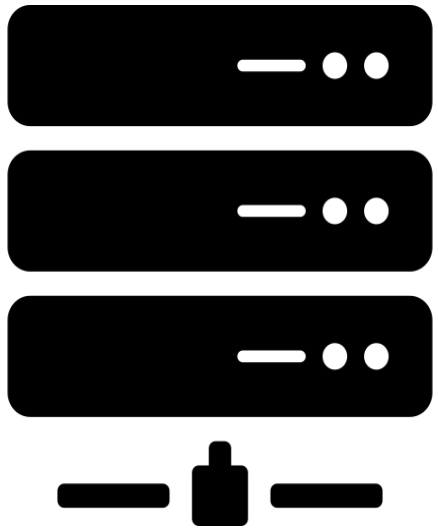


#7 시스템 도안

Web ⟷ Server

Web

- 사용자 입력 ID, PW
- Video, Image 파일
- Raspberry Pi 제어 정보



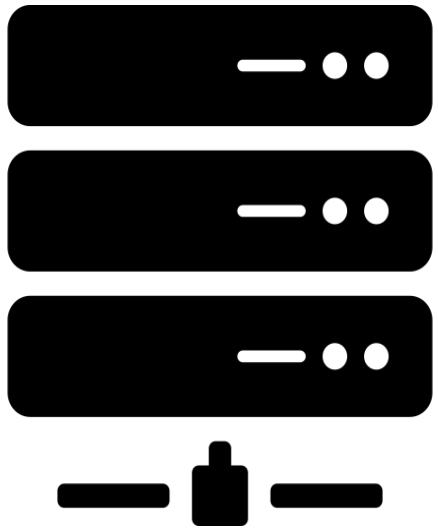
Server

#7 시스템 도안

Web ⟷ Server

Web

- 사용자 신원 확인
- Raspberry Pi 정보
- PlayList(Video, Image)



Server

#7 시스템 도안

Web ⟷ Server : 주요 소스 코드

```
//ServerThread -> rasp 접속확인 Thread
ServerThread server = new ServerThread();
Thread serverTh = new Thread(server);

//PingThread -> rasp 한테 10초마다 ping을 보내는 Thread
PingThread ping = new PingThread();
Thread pingTh = new Thread(ping);


//Thread 시작
serverTh.start();
pingTh.start();
```

- **serverTh**
  - ✓ Raspberry Pi 접속 관리
  - ✓ 동작에 따른 DB 갱신과 소켓통신
- **pingTh**
  - ✓ 10초마다 DB의 접속 관리 테이블에 등록되어 있는 IP목록에 ping을 보내 ON/OFF 확인

#7 시스템 도안

Web ⟷ Server : 주요 소스 코드

```
//USER 정보 수신
BufferedReader br = new BufferedReader(new InputStreamReader(client.getInputStream()));
StringBuilder sb = new StringBuilder();
String line;
while((line = br.readLine()) != null) {
    sb.append(line);
}
JSONObject obj = new JSONObject(sb.toString());

String userName = obj.get("user").toString();
String raspName = obj.get("rasp").toString();
String raspLat = obj.get("lat").toString();
String raspLng = obj.get("lng").toString();
String execContet = obj.get("exec").toString();
String category = obj.get("category").toString();

System.out.println("접속한 라즈베리파이 정보 > 소유주 : " + userName + ", 라즈베리파이명 : " + raspName + " 위도: " +raspLat + " 경도: " + ras
//MAIN 의 MAP 객체에 넣기
RaspInfo raspInfo = new RaspInfo(userName, raspName, raspLat, raspLng, execContet, category);
MainServer.activeList.put(ipv4, raspInfo);

//DB테이블 갱신
insertAccess(raspInfo, ipv4);
```

- Raspberry Pi가 ON 상태로 전환시 Server에 Ping을 보내고 Server는 그것을 기록
  - ▶ Server 기록 코드 중 일부

#7 시스템 도안

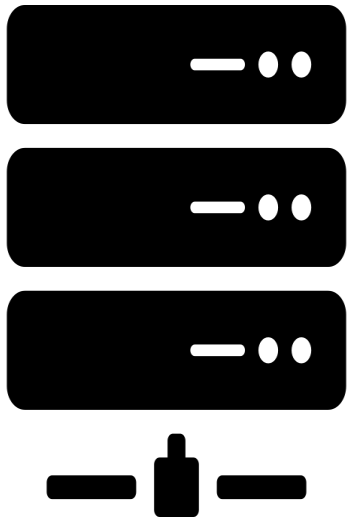
Web ⟷ Server : 주요 소스 코드

```
private void pingListCheck() { //IP주소 목록 DB에서 가져오기
    System.out.println("Ping 체크 시작");
    sql = "SELECT ipv4, raspname, username FROM accesslist;";
    try {
        pstmt = db.con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        while(rs.next()) { //핑체크 시작(라즈베리파이가 와이파이에 올바른 주소에 연결만 되어있으면 켜져있다고 판정)
            try {
                //IP주소만 가져오기
                InetAddress pingCheck = InetAddress.getByName(rs.getString("ipv4"));
                try {
                    if(!pingCheck.isReachable(2000)) {//2번 접속시도
                        //해당 ipv4 주소의 rasp에 Ping을 보내고 받을 수 없음.
                        //DB의 accesslist 테이블에서 행 삭제
                        sql = "DELETE FROM accesslist WHERE raspname=? AND ipv4=? AND username=?;";
                        pstmt = db.con.prepareStatement(sql);
                        pstmt.setString(1, rs.getString("raspname"));
                        pstmt.setString(2, rs.getString("ipv4"));
                        pstmt.setString(3, rs.getString("username"));
                        pstmt.executeUpdate();
                    };
                } catch (IOException e) {
                    System.out.println("pingListCheck()의 ping보내기 오류");
                }
            } catch (UnknownHostException e) {
                System.out.println("pingListCheck()의 ip가져오기 오류");
            }

        }

    } catch (SQLException e) {
        System.out.println("pingListCheck() 오류");
    }
```

- 10초 마다 접속 리스트 테이블에서 IP를 가져와 확인하고 관리하는 코드

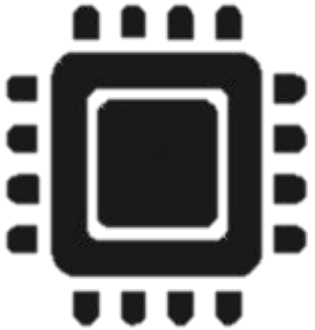
#7 시스템 도안

Server ⟷ Rasp



Server

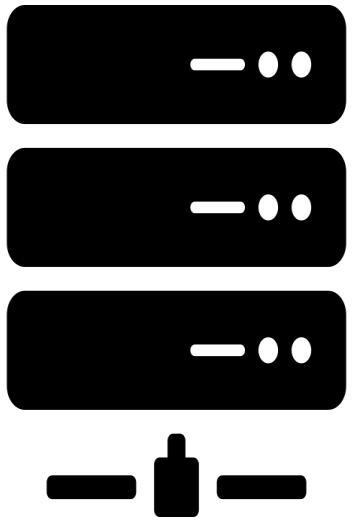
- 동영상 및 이미지
- 제어 정보
- Ping 으로 ON/OFF 확인



Raspberry Pi

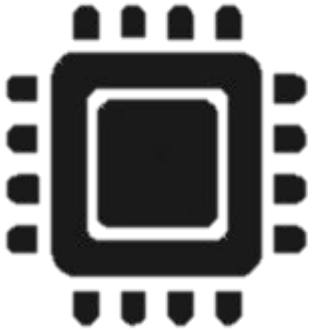
#7 시스템 도안

Server ⟷ Rasp



Server

- IP, 이름, 실행중인 파일
- 위치 정보
- ON/OFF 상태



Raspberry Pi

#7 시스템 도안

Server ⟷ Rasp : 주요 소스 코드

```
//파일 데이터 및 정보 읽기
String fileURL = "D:\\apache\\Apache24\\htdocs\\user\\"+userName+"\\"+category+"\\"+fileName+"."+extension;
File file = new File(fileURL);
fin = new FileInputStream(file);

fileSize = file.length();
byte[] buffer = new byte[DEFAULT_BUFFER_SIZE];
int readBytes;

//파일명과 카테고리 전송
JSONObject jsonOb = new JSONObject();
jsonOb.put("fileName", fileName);
jsonOb.put("category", category);
jsonOb.put("action", action);
String strJson = jsonOb.toString();



//ObjectOutputStream obOut = new ObjectOutputStream(out); //객체 직렬화
out.write(strJson.getBytes());



while( (readBytes = fin.read(buffer)) > 0) {
    out.write(buffer, 0, readBytes); //바이너리 데이터 전송
}
fin.close();

out.close();
socket.close();

//파일 전송 완료
System.out.println(ipv4+" 에 " +fileName+"."+extension+" 를 통신완료" + " 소유주 : "+userName);
```

- **WEB의 PlayList에서 동영상 교체 및 파일 보내기를 위한 코드**

```
$str = exec("java -jar ./sendToRasp5.jar ".$ipv4." ".$fileName." ".$fileExtention." ".$id);
```

▶ **WEB(PHP)에서 jar 파일을 실행 후 수행한다**

-

#7 시스템 도안

Server ⟷ Rasp : 주요 소스 코드

```
26
27 def readContentsList() :
28     videoList = []
29     imgList = []
30     gifList = []
31
32     #동영상
33     videoPath = '/home/pi/Desktop/Signage/display/video'
34     videoList = os.listdir(videoPath)
35     videoList.sort()
36
37     #이미지
38     imgPath = '/home/pi/Desktop/Signage/display/img'
39     imgList = os.listdir(imgPath)
40     imgList.sort()
41
42     #GIF
43     gifPath = '/home/pi/Desktop/Signage/display/gif'
44     gifList = os.listdir(gifPath)
45     gifList.sort()
46
47     return videoList, imgList, gifList
48
49
50
```

- **Raspberry Pi 내의 미디어 리스트를 읽어오는 코드**

#7 시스템 도안

Server ⟷ Rasp : 주요 소스 코드

```
def statusPrint():

    while True:
        global execContent, execCategory
        print(execContent + ' 실행중 ' + execCategory)
        time.sleep(5)


if __name__ == "__main__" :
    print("--SIGNAGE ON--")
    #라즈베리파이 켰을때 서버로 보내는 정보
    sendServer.sendRaspInfo()
    execContent, execCategory = sendServer.returnExecContent()
    print('시작화면 ' + execContent, execCategory)
    print("-------------------------")

    print(" 현재 저장된 콘텐츠 목록 읽어오기")

    videoList, imgList, gifList = readContentsList()
    print(videoList)
    print(imgList)
    print(gifList)

    if(execCategory == 'video') :
        path= "/home/pi/Desktop/Signage/display/video/"+execContent
        vDisplay.PlayNow(path)
    elif (execCategory == 'img') :
        iDisplay = DisplayImg.DisplayImg()
        iDisplay.SetPath(execContent)
        iDisplay.DisplayNow()
    print('-------------------------')

    #파일을 받는 메소드는 쓰레드에 넣어서 동작
    #화면을 제어하는 메소드도 쓰레드에 넣어서 동작
    #동시에 실행시켜야 하기 때문


    recvThread = threading.Thread(target=recvServer.getFileFromServer)
    #statusThread = threading.Thread(target=statusPrint)
    recvThread.start()
    #statusThread.start()


    recvThread.join() #Thread가 끝날때까지 MAIN 대기
    print("--SIGNAGE OFF--")
```

- Raspberry Pi 의 상태가 ON으로 전환시 수행할 코드.
- sendSever.sendRaspInfo()를 통해 정보를 송신

#7 시스템 도안

Server ⟷ Rasp : 주요 소스 코드

```
def sendRaspInfo() :
    global execContent, execCategory, RASPINFO, clientSocket, recvSocket
    try :
        #연결

        clientSocket.connect((HOST,PORT))

        #실행종이어야 하는 정보를 받아옴 (execContent, execCategory)
        recvSocket.bind((MYIP, MYPORT))
        recvSocket.listen(5)
        conn, addr = recvSocket.accept()

        #파일 받기
        data = 0
        try :
            data = conn.recv(1024)
        except Exception as e :
            print(e)
        print(data.decode('euc-kr'))
        de_data = json.loads(data.decode('euc-kr'))

        fileName = de_data['fileName']
        category = de_data['category']
        execContent = fileName
        execCategory = category

        RASPINFO['exec'] = execContent
        RASPINFO['category'] = execCategory

        #라즈베리파이 정보 전송
        #확인 및 json 만들기
        print(RASPINFO)
        RASPINFO = json.dumps(RASPINFO).encode("UTF-8")

        #전송
        clientSocket.sendall(RASPINFO)



        clientSocket.close()
        recvSocket.close()
        print("라즈베리파이 정보 전송 완료")
```

- **sendRaspInfo()**
  - ▶ **정보를 JSON 형태로 가공 후 송신하는 코드**

#7 시스템 도안

Server ⟷ Rasp : 주요 소스 코드

```
def getFileFromServer() :
    global execContent,execCategory
    execContent, execCategory = main.readContent()
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    soc.bind((HOST, PORT))
    soc.listen(5)
    print("파일 수신 준비 완료..." )

    #test
    #test = {'fileName':'img2', 'category':'img'}
    #en_da = json.dumps(test).encode('utf8')
    #test_da = json.loads(en_da.decode())

    while True:
        conn, addr = soc.accept()
        print('다음과 연결됨 : {}'.format(addr[0]))

        #파일 받기
        data = 0
        try :
            data = conn.recv(BUFSIZE)

        except Exception as e :
            print(e)
        de_data = json.loads(data.decode('euc-kr'))
        print(de_data)

        #파일명과 카테고리
        fileName = de_data['fileName']
        category = de_data['category']

        file = open('/home/pi/Desktop/Signage/display/'+category+'/'+fileName,
        try :
            while data:
                data = conn.recv(BUFSIZE)
                file.write(data)
            print(category + " 형식의 " +fileName + " 파일 수신 완료")
        except Exception as e:
                print(category + " 형식의 " +fileName + " 파일 수신 실패")
                print(e)
                soc.close()
        if(category == 'video'):
                path = "/home/pi/Desktop/Signage/display/video/"+fileName
                command = 'sudo killall -s 9 omxplayer.bin'
                os.system(command)
                time.sleep(1)
                vDisplay.PlayNow(path)
```

- **getFileFromServer()**
  - ▶ 정보를 JSON 형태로 수신 후 명령에 따른 기능을 수행하는 코드의 일부

#8 개선해야 할 점

1. VPN을 통한 외부망 접근

↳ 현재 같은 공유기을 사용해 내부망에서만 접근 가능

2. 스케줄링 기능 추가

↳ 날짜, 시간별로 따른 미디어 제어

3. HDD 저장 공간 증대

↳ 현재 1TB NAS용 HDD 사용 중

↳ 고화질 동영상 저장 및 사용자 증가에 따른 변화 필요

4. 세세한 Log 관리

↳ Server 모든 동작과 통신 기록을 위한 Log 관리 능력 필요.

Thank you

Q&A