Artisu 프로젝트

# 프로젝트 소개

Artisu는 NVIDIA의 Stygan 기술을 활용하여 사용자가 업로드한 인물 사진에 대해서 연령, 표정, 성별을 사용자가 요청한 대로 가공해주는 이미지 처리 서비스입니다.

# 코드 (Server)

|  |
| --- |
| host\_addr.sin\_family=AF\_INET;  host\_addr.sin\_port=htons(PORT);  host\_addr.sin\_addr.s\_addr=htonl(INADDR\_ANY);  memset(&(host\_addr.sin\_zero),0,8);  bind(socket\_fd,(struct sockaddr \*)&host\_addr,sizeof(struct sockaddr));  listen(socket\_fd,3);  size=sizeof(struct sockaddr\_in);  while(1){  printf("Start New TCP CONNECTION\n");  accepted\_fd=accept(socket\_fd,(struct sockaddr \*)&client\_addr,&size);  FILE\* file=NULL;  file=fopen("aligned\_images/man\_01.png","wb");  int file\_size=recv(accepted\_fd,&buffer,BUF\_SIZE,0);  file\_size=\*((int\*)buffer);  printf("file\_size = %d\n",file\_size);  send(accepted\_fd,"Recieve fils size\n\r",30,0);  recv\_length=0;  int total=0;  while(total<file\_size){  recv\_length=recv(accepted\_fd,buffer,BUF\_SIZE,0);  if(recv\_length>0){  total+=recv\_length;  printf("Target Download : %f%\n",((double)total/(double)file\_size)\*100);  if(recv\_length<BUF\_SIZE)  {  fwrite(&buffer,sizeof(char),recv\_length,file);  }  else  fwrite(&buffer,sizeof(char),BUF\_SIZE,file);}  }  send(accepted\_fd,"Server recieve Target file\n\r",40,0);  fclose(file);  //이하 생략 |

# 설명

# 안드로이드 클라이언트와 파일, 메시지를 송수신 받는 시스템을 구축하기 위해서 소켓프로그래밍을 이용하기로 하였습니다. 처음 소켓프로그래밍을 해보면서 TCP 통신이므로 서버와 클라이언트에서 보낸 모든 데이터가 제가 따로 신경 쓰지 않아도 송신만 제대로 한다면 모든 데이터를 정상적으로 수신받을 수 있을 것으로 생각하였습니다. 하지만 파일을 송수신해보면서 파일이 완벽하게 수신되지 않는 현상이 반복되었습니다. 무엇이 문제인지 고심하던 중, 한쪽에서 송신을 완료한 후 소켓 연결을 끊어버려 생기는 현상이라는 것을 알게 되었고, 파일 전송 시 파일 크기를 먼저 보내서 수신 측이 파일 크기만큼 파일을 정상적으로 수신하였을 때 확인 메시지를 송신 측에 보내주는 방법을 통해 문제를 해결하였습니다.

# 코드 (Android Client)

|  |
| --- |
| class TargetConnectThread extends Thread{  public void run(){  try{  InetAddress serverAddr = InetAddress.getByName("114.70.9.188");  Socket socket = new Socket(serverAddr,2020);  OutputStream os = socket.getOutputStream();  /\*  파일 송수신 코드  \*/  MSG="We are Done\n\r";  bytearr=MSG.getBytes();  os.write(bytearr);  os.flush();  dos.flush();//데이터 수신  //화면 출력  socket.close();  }catch(Exception e){  e.printStackTrace();  }  runOnUiThread(new Runnable() {  @Override  public void run() {  ImageView open=(ImageView)findViewById(R.id.imageView7);  open.setImageBitmap(x);  }  });  }  } |

# 설명

안드로이드 스튜디오를 이용하여 클라이언트 개발했습니다. 클라이언트 개발하면서 위에서 언급했던 것과 같이 소켓 프로그래밍을 통한 파일 송수신 작업 시 메시지를 통한 동기화가 필요하다는 것을 알게 되었습니다. 또한, 안드로이드에서 소켓을 사용하기 위해서는 메인 스레드를 사용해서는 안 되고 별도의 스레드를 활용해야 한다는 것을 배웠습니다.