

# 5월14일 회의록

날짜	2024.05.14
시간	19:30 ~ 21:00
수행자 이름	장진영
참여 인원	장진영, 강현준, 김호진, 이한빈
수행 내용	<p>1. 모든인원 GIT Repository 참여, 각자 코드 업로드 및 README파일 만들어서 다른 팀원이 볼 때 이해하기 편하게 만들기</p> <p>2. PNG 압축 파일을 decoding할 수 있는 알고리즘이 있는지 찾아봄, LZ77알고리즘이 파이썬으로 구현된 코드를 찾았고 C에서도 &lt;png.h&gt;헤더를 이용하여 작성할 수 있음을 알.</p> <p>하지만 keil에서 돌아갈지 모르기때문에 기존 프로젝트를 빨리 수행하고 다음주부터 PNG파일을 decoding해서 function 1, 2, 3을 구현하기로 했음.</p>
Trouble Shooting	<p>1. Memory relocation방법 2가지중에 RGBRGB형식으로 메모리 배치를 하게 되면 왜 안되는지 논의를 나눔</p> <p><a href="https://youtu.be/EFUYNofRHQI?si=DXMTfcMucINodsLZ">https://youtu.be/EFUYNofRHQI?si=DXMTfcMucINodsLZ</a> 영상을 참고하여 생각했을때 RGBRGB형식으로 메모리 배치를 하게된다면 파일 압축시 문제가 생길 것 같음[24분15초~] 참조.</p> <p>→ 압축 알고리즘 말고 다른 이슈가 있을 것 같은데 아직 발견하진 못함. ARM코드를 분석하면서 메모리 접근방식을 알아야 할 것 같다.</p> <p>2. Memory relocation을 통해서 최적화 코드를 짜고 기존 코드와 비교해봤는데 최적화한 코드의 실행시간이 더 길게 나왔다. 최적화 코드에서는 배열을 총 3번 돌기 때문에 load시간이 오래 걸리는건가 예상했는데 load시간은 기존 코드와 별 차이가 없었음. 압축할 때 더 오래 걸리는 것이라 판단</p> <p>→ 레지스터의 크기가 32bytes이기 때문에 기존 최적화 코드는 memory relocation만 진행하였는데 R이 저장된 곳에서 32bytes, B가 저장된 곳에서 32Bytes를 가지고와서 명령어수를 줄여보기로 함.</p>
다음 수행 계획	<p>5월 21일 까지 개인 보고서 제출, 보고서 필수 포함 내용 : 이미지 변환 함수의 동작 방식, 순서도, C, ARM코드(2개의 알고리즘 별 코드 각각 하나씩), 실행 후 memory map 사진, 두 코드 간의 성능 비교</p> <p>자세한 사항은 프로젝트 rev_2 Report 부분 참조하여 보고서 작성할 것</p>

