





목차

- 01 프로젝트 설명
- 02 프로젝트 결과
- 03 느낀점
- 04 추후 계획



01

프로젝트 설명



트러블슈팅 능력 향상



프로그래밍 능력 향상



대형 소프트웨어 분석 능력 갖추



Vmware에서 GUI를 어떻게 처리하는지?



01

프로젝트 설명

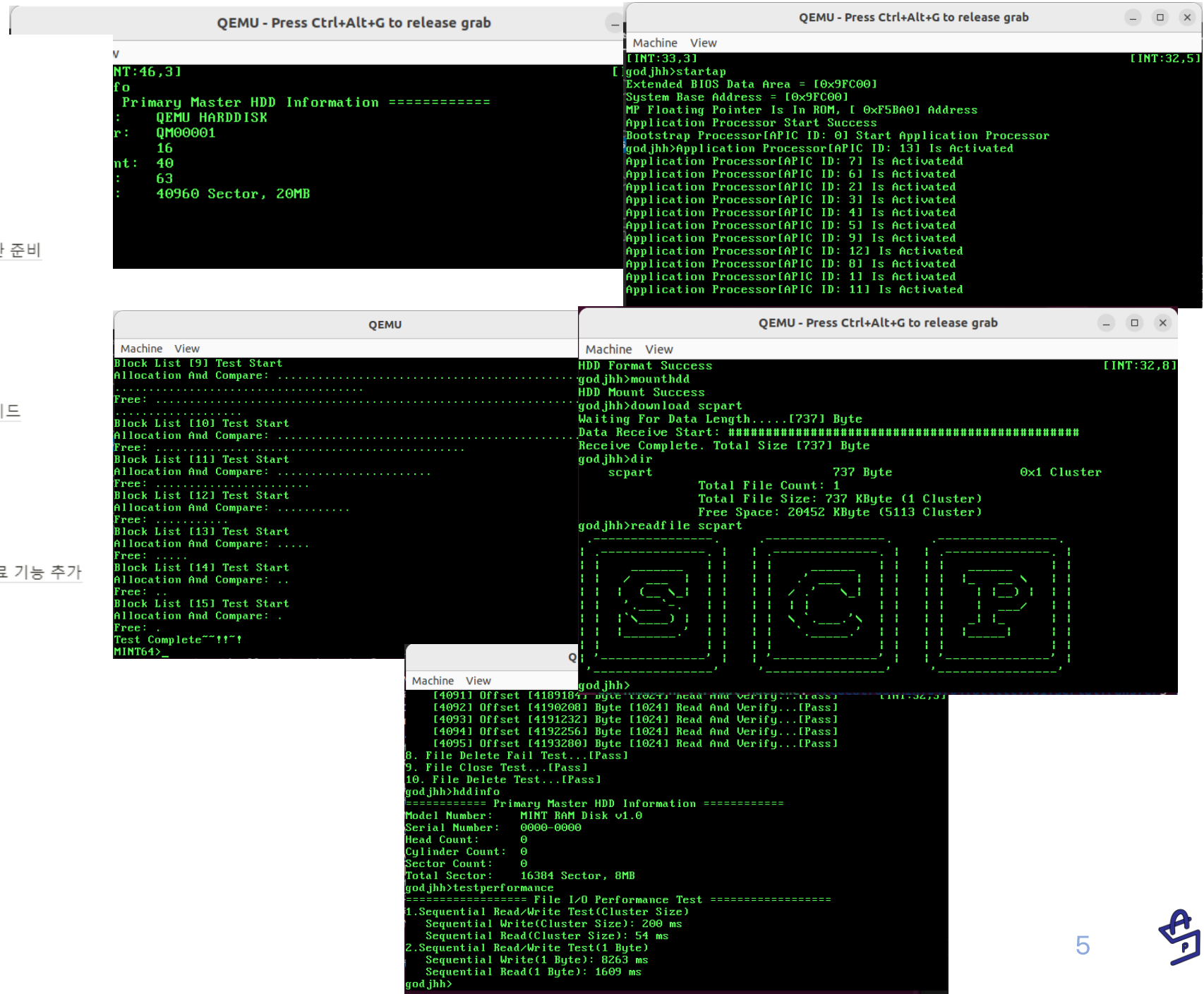
Weeks Tasks	1 Weeks	2 Weeks	3 Weeks	4 Weeks	5 Weeks	6 Weeks	7 Weeks
커널 페이징 활성화							
키보드 타이머 인터럽트							
멀티태스킹 멀티스레딩 실수 연산 기능							
메모리 관리 파일 시스템 시리얼 통신							
멀티코어 업그레이드							



01

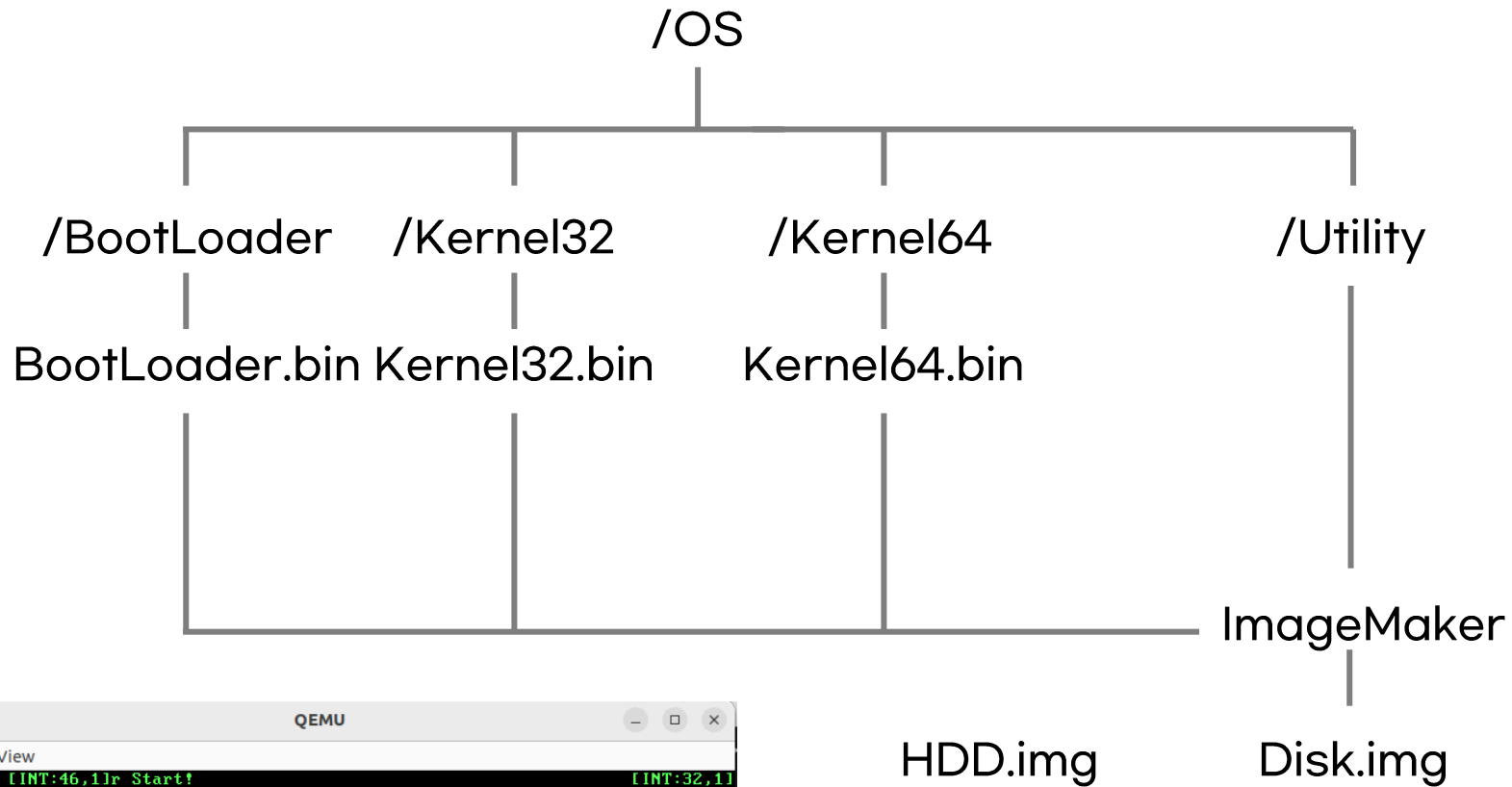
프로젝트 설명

- 개발 환경 구축
- 4장 부팅
- 5장 플로피 디스크에서 OS 이미지 로딩
- 6장 32비트 보호모드로 전환
- 7장 C언어로 커널 작성
- 8장 A20 게이트 활성화
- 9장 페이징 기능 활성화하여 64비트 전환 준비
- 10장 64비트 모드로 전환
- 11장 키보드 디바이스 드라이버 추가
- 12장 인터럽트
- 13장 PIC컨트롤러, 인터럽트 핸들러
- 14장 키보드 디바이스 드라이버 업그레이드
- 15장 콘솔 셀
- 16장 타이머 디바이스 드라이버
- 17장 멀티태스킹
- 18장 라운드 로빈 스케줄러
- 19장 멀티레벨 큐 스케줄러 + 태스크 종료 기능 추가
- 20장 태스크 동기화 문제 해결
- 21장 멀티스레딩
- 22장 실수 연산 기능
- 23장 메모리 할당
- 24장 하드디스크 드라이버
- 25장 파일시스템
- 26장 C표준 입출력 함수 추가
- 27장 캐시, 램 디스크 추가
- 28장 시리얼 포트 디바이스 드라이버
- 29장 코어 개수와 시스템 구성 정보 추출
- 30장 잠자는 코어 깨우기



02

프로젝트 결과



```
QEMU
Machine View
MINT64 OS [INT:46,1]r Start! [INT:32,1]
OS Image Loading... Complete!
Switch To Protected Mode Success!
Protected Mode C Language Kernel Start.....[Pass]
Minimum Memory Size Check.....[Pass]
IA-32e Kernel Area Initialize.....[Pass]
IA-32e Page Talbes Initialize.....[Pass]
Processor Vendor String.....[AuthenticAMD]
64bit Mode Support Check.....[Pass]
Copy IA-32e Kernel To 2M Address.....[Pass]
Switch To IA-32e Mode Success~!!
IA-32e C Language Kernel Start.....[Pass]
Initialize Console.....[Pass]
GDT Initialize And Switch For IA-32e Mode...[Pass]
TSS Segment Load.....[Pass]
IDT Initialize.....[Pass]
Total RAM Size Check.....[Pass], Size = 64 MB
TCB Pool And Scheduler Initialize.....[Pass]
Dynamic Memory Initialize.....[Pass]
Keyboard Activate And Queue Initialize.....[Pass]
PIC Controller And Interrupt Initialize.....[Pass]
File System Initialize.....[Pass]
Serial Port Initialize.....[Pass]
god.jhh>
```



02

프로젝트 결과

```
help          - Show Help
cls           - Clear Screen
totalram      - Show Total RAM Size
strtod        - String To Decial/Hex Convert
shutdown      - Shutdown And Reboot OS
settimer      - Set PIT Controller Counter0, ex)settimer 10(ms) 1(periodic)
wait          - Wait ms Using PIT, ex)wait 100(ms)
rdtsc         - Read Time Stamp Counter
cpuspeed      - Measure Processor Speed
date          - Show Date And Time
createtask    - Create Task, ex)createtask 1(type) 10(count)
changepriority - Change Task Priority, ex)changepriority 1(ID) 2(Priority)
tasklist      - Show Task List
killtask      - End Task, ex)killtask 1(ID) or 0xffffffff(All Task)
cpuload       - Show Processor Load
testmutex     - Test Mutex Function
testthread    - Test Thread And Process Function
showmatrix    - Show Matrix Screen
testpie       - Test PIE Calculation
dynamicmeminfo - Show Dynamic Memory Information
testsequalloc - Test Sequential Allocation & Free
Press any key to continue... ('q' is exit) :
```

```
[INT:33,5]    - Test Thread And Process Function
showmatrix    - Show Matrix Screen
testpie       - Test PIE Calculation
dynamicmeminfo - Show Dynamic Memory Information
testsequalloc - Test Sequential Allocation & Free
Press any key to continue... ('q' is exit) :
testranalloc  - Test Random Allocation & Free
hddinfo       - Show HDD Information
readsector    - Read HDD Sector, ex)readsector 0(LBA) 10(count)
writesector    - Write HDD Sector, ex)writesector 0(LBA) 10(count)
mounthdd      - Mount HDD
formathdd     - Format HDD
filesysteminfo - Show File System Information
createfile     - Create File, ex)createfile a.txt
deletefile     - Delete File, ex)deletefile a.txt
dir            - Show Directory
writefile      - Write Data To File, ex) writefile a.txt
readfile       - Read Data From File, ex) readfile a.txt
testfileio    - Test File I/O Function
testperformance - Test File Read/WritePerformance
flush          - Flush File System Cache
download       - Download Data From Serial, ex) download a.txt
showmpinfo     - Show MP Configuration Table Information
startap       - Start Application Processor
god.jhh>_
```



02

프로젝트 결과

```
godjhh>dir
Total File Count: 0
Total File Size: 0 KByte (0 Cluster)
Free Space: 20456 KByte (5114 Cluster)
```

```
godjhh>writefile scp.txt
scpscpscpscpcp
```

```
File Create Success
```

```
godjhh>dir
scp.txt          17 Byte          0x1 Cluster
Total File Count: 1
Total File Size: 17 KByte (1 Cluster)
Free Space: 20452 KByte (5113 Cluster)
```

```
godjhh>readfile scp.txt
scpscpscpscpcp
```

```
godjhh>deletefile scp.txt
File Delete Success
godjhh>dir
Total File Count: 0
Total File Size: 0 KByte (0 Cluster)
Free Space: 20456 KByte (5114 Cluster)
```

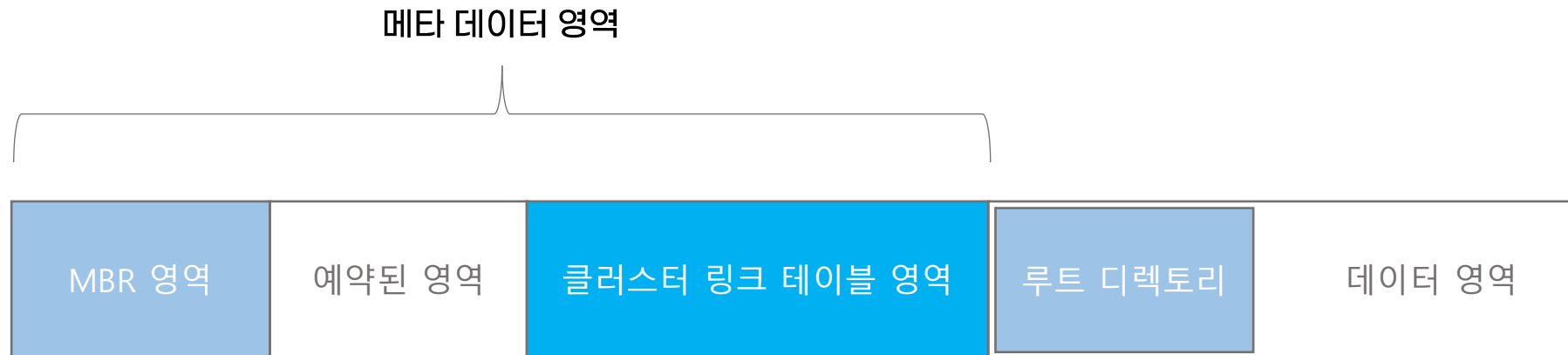


02

프로젝트 결과

파일 시스템이란?

데이터를 쉽고 빠르게 저장하고 검색할 수 있도록 저장 매체를 관리하는 일종의 규약



02

프로젝트 결과

MBR 영역

하드 디스크의 첫 부분에 위치하는 섹터로 부트 로더 코드, 파티션 정보, 파일 시스템 정보가 들어 있는 영역

예약된 영역

파일 시스템의 중요한 정보 저장하는 영역

클러스터 링크 테이블 영역

각 클러스터의 사용 여부와 다음에 연결된 클러스터의 인덱스 저장

데이터 영역

실제 데이터가 저장되는 영역



03

느낀점

기본기가 탄탄해야 한다.

아직 부족한게 많음

04

추후 계획

복습

꾸준하게 운영체제 공부

무조건 기초를 탄탄하게

리버싱 공부



