

1. List at least three different ways for structuring an operating system.
 - Simple structure
 - Layered structure
 - Microkernel
2. What technique do microkernels use to communicate between services?
 - message passing
3. Why do some systems store the operating system in firmware, while others store it on disk?

firmware를 알기 전에 ROM부터 알아야한다.

ROM은 초기화할 필요가 없으며 바이러스 걱정을 하지 않아도 된다.

OS를 ROM에 저장하는 것은 OS의 크기가 작거나 간단한 하드웨어를 지원하거나 험한 환경에서 실행되는 시스템에 적합하기 때문이다. 단, bootstrap code가 변경되면 ROM 칩 자체를 교체해야하는 단점이 있다.

ROM은 HW와 SW의 중간적 특성을 가지기 때문에 ROM의 모든 형태를 firmware라고 부른다.

firmware는 일반적으로 RAM에서 실행시킬 때보다 실행 속도가 느리다. 이를 보완하기 위해서는 firmware의 정보를 RAM으로 복사하여 실행하는 방법이 있다.

firmware는 disk대비 가격이 비싸기 때문에 용량이 크지 않다.

firmware에 저장하는 경우 기기의 부팅즉시 os를 사용할 수 있다. 그러나 다음의 단점 때문에 디스크에 저장한다.

디스크에 저장하게되면 OS의 용량을 firmware대비 더 늘릴 수 있다. 단위 용량대비 가격이 저렴하기 때문이다.

OS의 잦은 업데이트가 있다면 firmware보다 disk에 OS를 저장하는 것이 유리하다. firmware는 EPROM이 있다고 해도 데이터를 수정하기가 어렵기 때문이다. 수정이 되더라도 EPROM은 UVlight를 쬌어서 데이터를 지워야하는데, 일반인이 이 작업을 하기는 불가능하다. EEPROM이 들어온다고 해도 DISK방식이 속도가 더 빠르다. 특히 SSD는 더욱 빠르다.

그렇기 때문에 요즘 컴퓨터는 EEPROM(ROM이나 EPROM을 사용하기는 한다.)에 BIOS를 저장하고, 대부분의 OS 정보는 DISK에 저장하여 부팅한다.

4. What is the name of the small piece of code that locates the kernel and loads it into mainmemory?

- bootstrap

5.

5-1. What is the main advantage of the microkernel approach to system design?

5-2. How do userprograms and system services interact in a microkernel architecture?

5-3. What are the disadvantages of using the microkernel approach?

5-1. microkernel의 주요 장점은 OS 확장성이 좋다는 것이다. 새 서비스는 모두 user space에 추가되기 때문에 kernel을 변경하지 않아도 된다. 더욱이 user process로 프로그램이 실행되기 때문에 보안과 신뢰성이 높다.

5-2. user programs과 system service는 message passing과 같은 통신 메커니즘을 사용하여 micro kernel architecture에서 interact한다.

5-3.

- micro kernel system에서 service를 제공하는 것은 일반적인 단일 시스템에 비해서 성능이 낮고 구성 비용이 더 들어간다.

- 드라이버가 실행될 때는 내용의 변경이나 함수 호출이 필요하다.

6. What are the advantages of using loadable kernel modules?

Device driver를 kernel에 하드코딩하지 않아도 된다.

7. (프로그램 작성 및 strace 사용) In Section 2.3, we described a program that copies the contents of one file to a destination file. This program works by first prompting the user for the name of the source and destination files. Write this program in Linux. Be sure to include all necessary error checking, including ensuring that the source file exists.

Once you have correctly designed and tested the program, if you used a system that supports it, run the program using a utility that traces system calls. Linux systems provide the strace utility

파일은

u15165@solgae:~\$fileCopy.c 에 올려두었습니다.

solgae.yonseil.ac.kr - u15165@solgae: ~ - VT

메뉴(F) 수정(E) 설정(S) 제어(C) 창(W) 도움말(H)

```
u15165@solgae:~$ cc fileCopy.c
u15165@solgae:~$ ./a.out
Enter name of source file: u15165-1-2
Enter name of destination file: test
files are succesfully copied.u15165@solgae:~$ strace a.out
strace: Can't stat 'a.out': No such file or directory
u15165@solgae:~$ strace test
execve("/usr/bin/test", ["test"], 0x7ffdc50a77c0 /* 27 vars */) = 0
brk(NULL)                               = 0x56162249a000
arch_prctl(0x3001 /* ARCH_??? */ , 0x7ffcd521bf60) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=79142, ...}) = 0
mmap(NULL, 79142, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f4641821000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0\0\1\0\0\0\360A\2\0\0\0\0"... , 832) = 832
pread64(3, "\6\0\0\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\3\0\0\0GNU\0\237\333t\347\262\27\320\1\223\27*\202C\370T\177"... , 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029560, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f464181f000
pread64(3, "\6\0\0\4\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32,
```

solgae.yonseil.ac.kr - u15165@solgae: ~ - VT

메뉴(F) 수정(E) 설정(S) 제어(C) 창(W) 도움말(H)

```
WRITE, 3, 0x1e7000) = 0x7f4641815000
mmap(0x7f464181b000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f464181b000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f4641820580) = 0
mprotect(0x7f4641815000, 16384, PROT_READ) = 0
mprotect(0x561621b74000, 4096, PROT_READ) = 0
mprotect(0x7f4641862000, 4096, PROT_READ) = 0
munmap(0x7f4641821000, 79142)            = 0
brk(NULL)                               = 0x56162249a000
brk(0x5616224bb000)                     = 0x5616224bb000
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=7031072, ...}) = 0
mmap(NULL, 7031072, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f4640f78000
close(3)                                = 0
close(1)                                = 0
close(2)                                = 0
exit_group(1)                           = ?
+++ exited with 1 +++
u15165@solgae:~$
```