Day1

BIOS

Bootloader

Kernel

Day2

**Object File**

**(https://docs.oracle.com/cd/E19683-01/817-3677/chapter6-46512/index.html)**

Object files participate in both program linking and program execution

1. Relocatable file

* 코드와 데이터를 포함한 세션을 hold.
* Executable files, shared object files, or another relocatable object 만들기 위해 다른 오브젝트 파일과 링크 됨.

1. Executable file

* 실행 준비가 된 프로그램, exec(2)가 프로그램의 프로세스 이미지를 어떻게 만드는지 구체화

1. Shared object file
2. Link-editor

the link-editor can process this file with other relocatable and shared object files to create other object files.

1. Runtime linker

the runtime linker combines this file with a dynamic executable file and other shared objects to create a process

An ELF header resides at the beginning of an object file and holds a **road map** describing the file's organization.

**Sections**

ELF 파일에서 처리될 수 있는 눈에 보이지 않는 가장 작은 유닛

An object file's section header table lets one locate all the file's sections. The section header table is an array of Elf32\_Shdr structures as described below. A section header table index is a subscript into this array. The ELF header's e\_shoff member gives the byte offset from the beginning of the file to the section header table; e\_shnum tells how many entries the section header table contains; e\_shentsize gives the size in bytes of each entry. Some section header table indexes are reserved; an object file will not have sections for these special indexes.

**Segments**

Exec(2) 또는 runtime linker 에 의해 메모리 이미지에 매핑될 수 있는 최소한의 유닛으로 SectEions 들의 집

**Program header**

A program header table, if present, tells the system how to create a process image. Files used to generate a process image, executables and shared objects, must have a program header table; relocatable objects do not need such a table.

**Section Header**

A section header table contains information describing the file's sections. Every section has an entry in the table. Each entry gives information such as the section name, the section size, and so forth. Files used in link-editing must have a section header table; other object files might or might not have o모든시Sene.

**ELF (**[**https://refspecs.linuxfoundation.org/elf/elf.pdf**](https://refspecs.linuxfoundation.org/elf/elf.pdf)**)**

**Program Header**

An executable or shared object file's program header table is an array of structures, each describing a segment or other information the system needs to prepare the program for execution. An object file segment contains one or more sections. Program headers are meaningful only for executable and shared object files. A file specifies its own program header size with the ELF header's e\_phentsize and e\_phnum members

**ELF Header**

Some object file control structures can grow, because the ELF header contains their actual sizes. If the object file format changes, a program may encounter control structures that are larger or smaller than expected. Programs might therefore ignore "extra" information. The treatment of "missing" information depends on context and will be specified when and if extensions are defined.