(슬라이드 1)   
안녕하십니까 오늘 리눅스 booting과정에 대해 발표하게 된 소프트웨어학과 22학번 이이슬입니다.

(슬라이드 2)   
전통적인 리눅스의 부팅 과정에 대해 보겠습니다.  
먼저 컴퓨터의 전원을 켜게 되면 Power On Self Test, POST라는 과정을 거칩니다. 이거는 리눅스가 수행하는 과정이 아니라 컴퓨터가 하는 과정입니다. 만약에 하드웨어가 잘못되어 있다면 부팅이 불가능 하겠죠, 그래서 컴퓨터 자신이 CPU, 메모리, 하드디스크 등 하드웨어가 제대로 동작하나 점검을 하는 과정입니다.  
이제 점검을 하고 다음으로 BIOS라고 하는 베이직 입출력 시스템이 동작하는 과정이고 이 친구가 하드디스크에 부팅장치가 있는지 USB에 부팅장치가 있는지 부팅 장치를 검색합니다.  
이제 부팅 장치가 검색이 되었다면 리눅스 시스템 같은 경우에는 일반적으로 사용되는 그럽(grub)이라고 하는 부트 로더가 실행이 되고, 이 부트로더가 실행이 되면 실행할 커널의 목록을 보여주게 되는데요.   
(슬라이드 3으로 이동) 슬라이드 3을 보시면 그럽(grub)이 실행된 상태로, 실행할 커널의 목록을 보여주는 것을 확인할 수 있습니다.  
(슬라이드 2로 이동) 이제 실행할 특정 커널을 선택해 주면 커널이 메모리에 올라가게 되고, PID(피아이디) 1번 프로세스가 실행되게 됩니다. 이는 예전 리눅스 같은 경우에는 init(이닛)이라고 하는 프로세스였고요, 요즘 리눅스에서는 systemd(시스템D)라는 프로세스입니다.  
이 PID 1번 프로세스가 대부분의 리눅스 시스템이 동작하는데 필요한 기타 필수 프로그램들을 실행시켜 주게 됩니다. 그러면 이제 부팅이 완료가 되는 거죠.

(슬라이드 3 넘어가고)

(슬라이드 4)   
BIOS, 베이직 입출력 시스템 같은 경우에 이와 같이 하드웨어 관련된 설정이 있는데 앞서 BIOS가 부팅 장치를 검색한다고 했는데 표시되어 있는 Boot 메뉴에 가보시면 부팅 장치의 검색 순서 같은 것을 설정할 수 있습니다.   
만약에 윈도우나 리눅스를 포맷이나 부팅을 할 때 USB를 꽂아서 한다면 Boot 메뉴에 가서 USB를 먼저 검색하도록 설정하는 것이죠.

(슬라이드 5) 이제 부팅 장치가 검색이 돼서, 리눅스 같은 경우에는 grub이라고 하는 애가 실행이 된다고 하였는데 이 grub의 설정파일을 가보면 이와 같이 나오는데요. (슬라이드 3으로 이동) 여기에서 커널의 목록을 보고 다시 돌아가보면 (슬라이드 5로 이동) 제가 표시해둔 것과 동일 한 것을 볼 수 있죠.  
그러니까 이 grub.cfg라고 하는 설정파일에 아까 선택할 수 있던 커널들이 추가가 되어 있다는 것입니다. 만약에 커널 버전을 업그레이드하고 싶다면 새로운 커널을 다운로드 받고, 이런 설정파일에 추가를 해주면 되는 거겠죠.

(슬라이드 6) 이제 부팅 될 때 보면, 중간에 메시지들이 많이 뜨는데 이를 다 보지 못하죠.  
이때, var 밑에 log 밑에 boot.log라는 파일을 보면 부팅됐던 동안 출력됐던 메시지들을 이와 같이 확인할 수 있습니다.   
만약에 부팅이 된 거는 같은데 정상적으로 부팅이 된 거 같지 않을 때 이런 로그 파일을 확인해보고 잘못된 걸 바로잡을 수 있습니다.

(슬라이드 7) 다음으로는 런 레벨이라는 것을 볼 텐데요.   
런 레벨이란 리눅스가 부팅할 때 시스템의 상태를 나타내는 값입니다.  
아까 부팅할 때 기타 필수적인 프로그램들을 실행한다 하였는데, 리눅스는 기타 필수 프로그램들을 런 레벨에 맞는 프로그램들만 실행시켜 주는 것입니다.  
런 레벨이 0번부터 6번까지 있는데 예를 들어 3번 런 레벨로 시스템이 실행되었다면 3번 런 레벨에서 실행되야 하는 맞는 프로그램들만 실행되는 것이죠.  
즉, 기타 필수 프로그램이 실행된다는 것은 런 레벨에 맞는 기타 필수 프로그램들만 실행이 된다는 의미입니다.

(슬라이드 8) 런 레벨 관련 명령어를 보면 who -r이 있는데 이는 현재 및 이전 런 레벨을 확인하는 명령어이고, init은 런 레벨 스크립트를 실행하는 명령어이고 etc 파일에서 rc로 시작하는 디렉토리들을 보면 rc0부터 6까지 런 레벨에 맞는 런 레벨 스크립트를 확인할 수 있고 디렉토리로 이동해서 리스트를 보면 K로 시작하는 것은 이 런 레벨에서 프로그램을 끄는 것, S로 시작하는 것은 이 런 레벨에서 프로그램을 실행하는 것입니다.

(슬라이드 9) 앞서 PID 프로세스가 실행될 때 Init 프로세스와 Systemd 프로세스에 대해 말했는데 2개에 대해 알아보겠습니다. 먼저 바로 전에 root 권한 명령어로도 사용 가능했던 Init을 먼저 보겠습니다.  
커널이 메모리로 로드되면 Init 프로세스가 시작되는데요. Init 프로세스는 리눅스 시스템의 부팅 및 초기화를 담당하는 핵심 프로세스입니다.  
이 단계에서 Init은 다음과 같이 Init 또는 Systemd 초기화, 파일 시스템 마운트, 시스템 서비스 시작, 로그인 프롬프트 또는 GUI 표시하는 작업을 수행합니다.  
아래와 같이 booting 했을 때 보는 장면이 바로 커널로드가 되면서 init 프로그램이 실행되는 거라고 할 수 있습니다.

(슬라이드 10) 다음으로는 Systemd 프로세스입니다.   
Systemd는 리눅스 시스템 초기화 및 관리 프로세스로, init시스템의 대체물로 도입되었으며 서비스 관리, 디바이스 관리, 로깅, 프로세스 관리 등 다양한 시스템 관리 작업을 수행하는데 사용됩니다.  
그리고 주요 특징으로는 다음과 같이 병렬 부팅, 유닛 기반 관리, 로그 및 저널링, 타이머와 스케줄링, 저전력 관리가 있습니다.

(슬라이드 11) 이제 마지막으로 Init 0~5와 systemd를 비교하겠습니다.  
init 0은 시스템 종료를 위한 레벨이며, systemd의 poweroff.target은 동일한 역할을 합니다.  
다음으로 init 1은 단일 사용자 모드로, systemd의 rescue.target도 이와 유사한 역할을 합니다.  
init2는 네트워킹을 비활성화한 다중 사용자 모드이며 systemd의 multi-user.target은 비슷한 역할을 하고,  
init3은 init2와 동일하나 네트워크가 활성화 된 모드이며, systemd는 위와 동일하게 multi-user.target이고, init3과 동일한 역할을 합니다.  
init 4는 사용되지 않는 레벨이나 systemd에서는 사용자정의 타깃을 정의할 수 있고요.  
마지막으로 init5는 그래픽 환경을 지원하는 다중 사용자 모드로, systemd의 graphical.target은 크래픽 환경을 활성화 합니다.

(슬라이드 12) 더 궁금한 점이 있다면 자료를 참고해 주시고, 이상으로 발표를 마치겠습니다.