**운영체제**

금(12) 최상호 교수님

**Assignment1**

**컴퓨터정보공학부**

**2018202076**

**이연걸**

* **Introduction**

이번 과제는 VMware에 Ubuntu를 과제에서 요구하는 버전, 사양에 맞게 설치하고 과제에서 요구한 리눅스 명령어를 사용해 디렉토리 생성, 파일 생성, 파일 복사, 권한 설정, 파일 출력 등을 수행하는 것이었다. 그리고 kernel을 다운로드하고 커널 컴파일의 모든 과정을 터미널과 vi를 사용해 작성하는 것이었다. 마지막으로 ctags와 cscope를 사용해 Linux agp…이 실행되는 지점을 찾고 커널의 부팅 메시지가 문제에서 제시된 조건에 맞게 출력되도록 소스코드를 수정하는 것이다.

* **Conclusion & Analysis**

1. Assignment 1-1

리눅스 설치

텍스트이(가) 표시된 사진

자동 생성된 설명

리눅스 명령어

텍스트이(가) 표시된 사진

자동 생성된 설명

1. Assignment 1-2

Kernel Source 다운로드

텍스트이(가) 표시된 사진

자동 생성된 설명

$ sudo su

관리자 계정으로 전환하게 해준다.

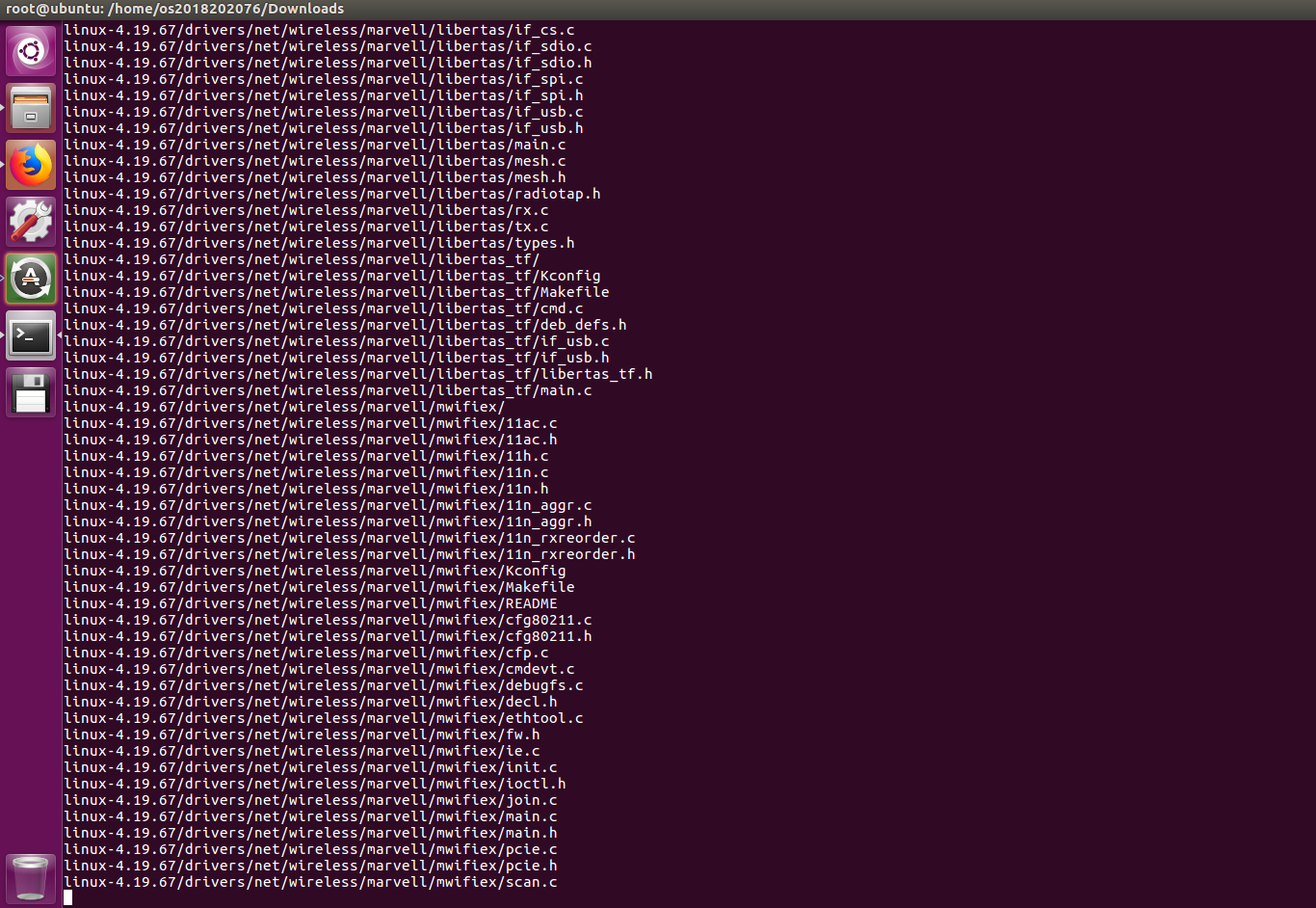
텍스트이(가) 표시된 사진

자동 생성된 설명

$ sudo wget http://cdn.kernel ...

전달된 url에서 파일 다운로드를 가능하게 해준다.

Kernel Source 압축 해제



텍스트, 전자기기이(가) 표시된 사진

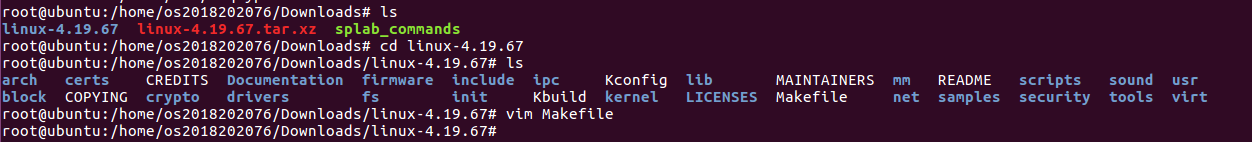
자동 생성된 설명

$ tar -Jxvf li …

여러 개의 파일을 하나로 묶거나 풀 때 사용한다.

-를 사용해 옵션을 붙였는데 앞에서부터 xz관련 옵션 명시, tar파일 풀기, 실행 대상의 파일 내용 명시, 작성 대상이 되는 tar 파일의 이름 지정이다.

Kernel Extra Version 수정



텍스트이(가) 표시된 사진

자동 생성된 설명

Vi editor를 사용해 Makefile의 설정을 변경한다.

Kernel 환경 설정 다운로드

텍스트이(가) 표시된 사진

자동 생성된 설명

$ sudo apt-get install build-ess …

Apt-get 패키지 관리자를 사용해 필요한 패키지들을 다운받는다.

Kernel 환경 설정 $ make menuconfig

텍스트이(가) 표시된 사진

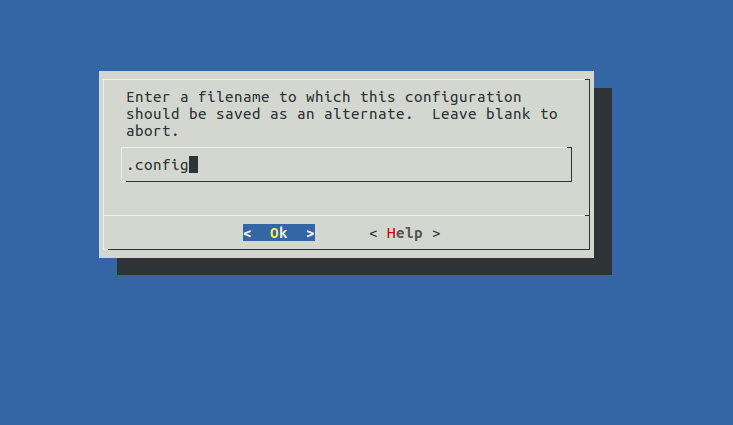
자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명



텍스트이(가) 표시된 사진

자동 생성된 설명

$ make menuconfig

커널 환경 설정을 해준다.

Kernel Compile

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

$ make -j12

컴파일을 12개의 스레드에 분할해서 실행한다.

Module 설치

텍스트이(가) 표시된 사진

자동 생성된 설명

$ make modules\_install

Makefile의 modules\_install 규칙을 실행한다.

Compile된 Kernel을 Boot Loader에 등록

텍스트이(가) 표시된 사진

자동 생성된 설명

$ make install

/arch/x86/boot/ 경로에 있는 bzImage라는 이름의 Kernel image를 /boot에 복사하고 System Map(System.map)을 /boot로 복사한다 .그리고 Grub 부트로더에 자동으로 등록해준다.

이 세가지 기능을 가능하게 하는 install 규칙을 실행한다.

Grub 설정

텍스트이(가) 표시된 사진

자동 생성된 설명

재부팅 후 커널 버전 확인

텍스트, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

$ uname -r

uname은 시스템 정보를 출력해준다. -r 옵션으로 커널의 릴리즈 정보를 출력했다.

-----------------------------------------------------------------------------------------

1. Assignment 1-3

**리눅스 커널 코드에서 수정한 부분 명시**

텍스트이(가) 표시된 사진

자동 생성된 설명

Linux agp…이 실행되는 지점은 agp\_init()이다. 그렇기 때문에 과제에서 요구한 조건대로 Linux agp…이 실행되는 지점에 커널 메시지가 과제에서 요구한 조건에 부합하게 “Linux apart interface v%d.%d\n” 앞에 os2018202076\_을 추가해 주었다.

그리고 Linux의 agp를 실행시키는 함수의 함수명인 agp\_init과 argument의 값인 void를 출력하도록 커널 코드를 수정했다.

수정한 코드는 다음과 같다.

printk(KERN\_INFO “os2018202076\_arg in agp\_init (void)\n”);

**소스코드 경로 작성**

텍스트이(가) 표시된 사진

자동 생성된 설명

더 정확한 경로는 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

/home/os2018202076/Downloads/linux-4.19.67/drivers/char/agp/backend.c

**검색한 캡쳐 화면 첨부**

텍스트이(가) 표시된 사진

자동 생성된 설명

**결과 화면 캡쳐**

텍스트이(가) 표시된 사진

자동 생성된 설명

* **고찰**

이번 과제에선 가상 머신에 Linux를 설치하고 명령어를 사용해 파일, 디렉토리 생성, 수정, 삭제 권한 변경, 커널 컴파일, start\_kernel의 이해 등을 요구했다.

과제를 진행하다가 의문사항이 한가지 있었다. 과제 조건에 “가상 머신에 할당한 CPU core 수의 1.5배 ~ 2배 정도면 적합” 이라는 말이 쓰여 있었다. 왜 thread의 수가 CPU 코어 수의 2배인 것일까? 그에 대한 해답은 하이퍼스레딩이었다.

하이퍼 스레딩은 1코어를 가진 CPU 프로세서를 논리적인 2코어로 만드는 기술이다. 커널 컴파일과 같은 처리할 데이터가 많은 작업을 진행할 때 프로세서의 처리량을 높여 성능을 높이는 기술이다.

내 노트부에 설치된 CPU도 하이퍼스레딩을 지원하는지 여부가 궁금해서 확인해 보았다.

테이블이(가) 표시된 사진

자동 생성된 설명

작업 관리자의 성능 탭의 CPU를 코어와 논리 프로세서가 나뉘어져 있다. 코어 개수는 8개, 논리 프로세서 개수는 16개로 적혀 있다. 하이퍼 스레딩을 지원하는 CPU 프로세서는 코어 수와 논리 프로세서 수가 다르게 표시되기에 현재 CPU 프로세서가 하이퍼스레딩을 지원함을 확인할 수 있었다.

실수했던 부분도 있었는데 ctags, cscope은 Man Page를 참고해 어렵지 않게 사용할 수 있었지만 과제의 조건이었던 “본인이 기술한 start\_kernel()에서 Linux agp…이 실행되는 지점에 기반하여 수정한 코드”를 제대로 이해하지 못했다.

리눅스의 부팅 과정 중 GRUB를 거쳐 Kernel단계에 들어가면 start\_kernel()이 실행되는데 Linux\_agp이 실행되는 지점은 start\_kernel이 아닌 agp\_init이다. 처음에 start\_kernel() 함수에 두 문구를 출력하는 printk문을 넣어서 두 문구가 중복되어 출력되었는데 agp\_init함수로 옮겨서 해결할 수 있었다.

* **Reference**

-https://jnstory.net/2462#:~:text=%ED%95%98%EC%9D%B4%ED%8D%BC%EC%8A%A4%EB%A0%88%EB%94%A9%20%ED%99%95%EC%9D%B8%20%EB%B0%A9%EB%B2%95,%EC%96%B4%EB%A0%B5%EC%A7%80%20%EC%95%8A%EA%B2%8C%20%ED%99%95%EC%9D%B8%ED%95%A0%20%EC%88%98%20%EC%9E%88%EC%8A%B5%EB%8B%88%EB%8B%A4.