

2024년 2학기 운영체제실습

Assignment 1

System Software Laboratory

School of Computer and Information Engineering

Kwangwoon Univ.

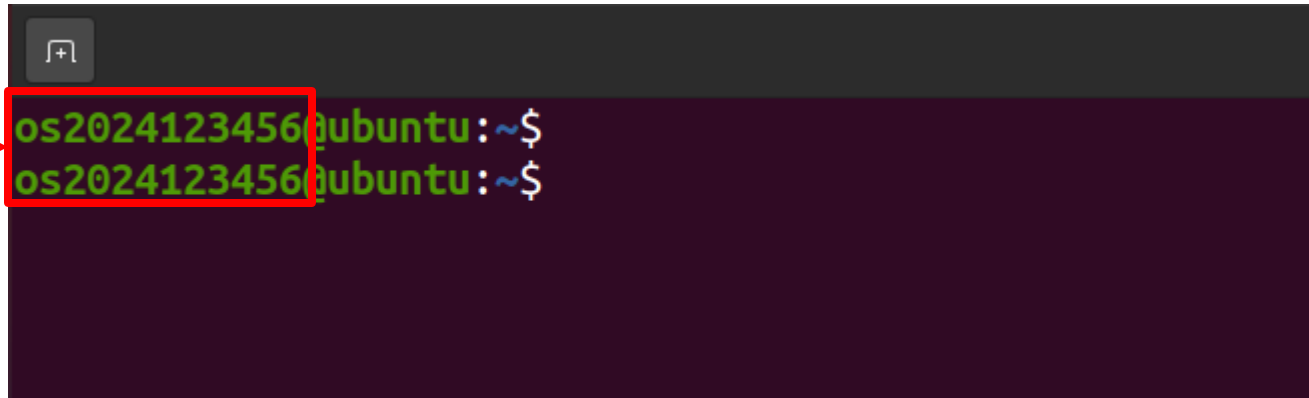
Contents

- **Assignment1-1**
 - Linux installation
- **Assignment1-2**
 - Linux Kernel Compile
- **Assignment1-3**
 - Practice programming tools
- **Appendix**
 - Appendix A. printk()
 - Appendix B. dmesg command
 - Appendix C. tar.xz compress

Assignment 1-1

- Linux Installation

- 리눅스를 설치하는 과정을 캡처하고, 이를 짧은 설명과 함께 보고서에 첨부
- 과제 요구 사항
 - 리눅스 설치 과정을 간단하게 캡처 및 설명
 - 가상 머신에서 설치하는 경우, 가상 머신을 만드는 과정도 포함
 - 계정 ID는 “os학번”으로 할 것 (e.g. os2024123456)

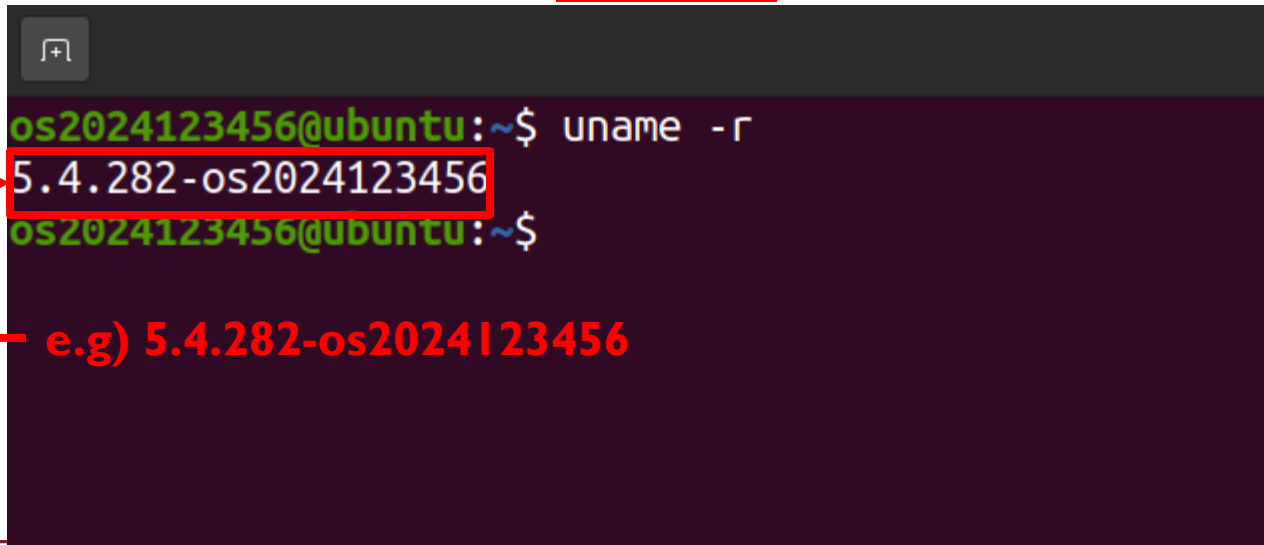


```
os2024123456@ubuntu:~$  
os2024123456@ubuntu:~$
```

Assignment 1-2

Kernel 5.4.282 Compile

- 우분투 내에서 Kernel을 다운로드 하고, 컴파일의 모든 과정을 terminal과 vi를 이용하여 보고서에 첨부
- 사용한 명령어를 순서대로 보고서에 캡처하여 작성
- 과제 요구 사항
 - 커널 컴파일 과정을 terminal과 vi를 사용하여 캡처
 - 각 명령어가 어떠한 기능을 하였는지 간단히 서술
 - 5.4.282 커널로 재 부팅 후 버전 확인
 - “uname -r”의 결과가 “5.4.282-os본인 학번”이 나오도록 진행할 것



```
os2024123456@ubuntu:~$ uname -r
5.4.282-os2024123456
os2024123456@ubuntu:~$
```

e.g) 5.4.282-os2024123456

Assignment 1-3

- Practice programming tools(Ctags , Cscope)
 - Ctags, Cscope를 활용하여 해당 함수 및 파일을 찾아 수정
- Linux agpgart interface가 실행되는 지점에서 Linux Kernel Message 출력
 - Linux agpgart가 실행되는 지점에 커널 메시지가 다음과 같이 출력되도록 커널 코드 수정-----**(1)**

```
printk(KERN_INFO "Linux agpgart interface v%d.%d\n",
```
 - Linux agpgart를 실행시키는 함수의 함수명과 argument의 값을 출력하도록 커널 코드 수정 -----**(2)**

[출력 예시]

os2024123456_Linux agpgart interface v0.103

```
os2019110613@ubuntu:~$ dmesg | grep "os2019110613" -n
1404:[    5.546304] os2019110613_Linux agpgart interface v0.103
1405:[    5.546304] os2019110613_arg in FUNCTION ( arg )
```

↑
argument

Assignment 1-3(Cont'd)

- 과제 요구사항

- Solution

- 커널 코드 내에 **printk()** 함수를 사용한 코드를 추가하여 과제 수행
 - 로그 레벨은 **KERN_INFO** 사용 (강의 자료 “Appendix A” 참고)
 - 출력 메시지는 5 Page의 [출력 예시] 참고.

- 확인 방법

- 부팅 직후 **dmesg**, **grep**으로 확인 시 (1), (2) 메시지가 **연달아** 출력되어야 한다. (“Appendix B” 참고)
 - 주의사항 1, 2를 참고하세요. (7page)

- 보고서에 다음의 내용을 필히 포함

- 리눅스 커널 코드에서 수정한 부분을 명시 (소스코드 path도 포함)
 - 수정한 위치가 **정당한 이유**를 명시 (e.g. 검색한 캡처 화면 첨부, ...)
 - 결과 화면 캡처

- Softcopy 제출 시, 수정한 커널 코드 파일 포함

- **init/main.c**를 수정했다면,
main.c 파일을 첨부하여 제출하고, 수정한 소스코드 path를 보고서에 명시.

Assignment 1-3(Cont'd)

주의 사항 1

- 프린트 되는 문자열은 중복되어 여러 번 나오면 안됩니다.

[ex]

```
1010: [      4.107586] os2024123456_Linux agpgart interface v0.103
1011: [      4.107590] os2024123456_arg in function_name ( argument )
...
1404: [      5.107586] os2024123456_Linux agpgart interface v0.103
1405: [      5.107586] os2024123456_arg in function_name ( argument )
```

- 아래와 같이 **중복 없이** 한 set로 나와야 인정합니다.

두 문구 중 어느 하나라도 중복으로 출력되면 감점입니다.

[ex]

```
1404: [      5.107586] os2024123456_Linux agpgart interface v0.103
1404: [      5.107586] os2024123456_arg in function_name ( argument )
```

주의 사항 2

- 제대로 된 위치**에서 문자열을 출력해야 인정합니다.
- 본인이 기술한 “start_kernel()에서 Linux agp...이 실행되는 지점”에 기반하여 수정한 코드만 인정합니다.

Report Requirements

- **Ubuntu 20.04.6 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
 - 과제 이름 → Assignment #1
 - **과제 내용**
 - Introduction
 - 과제 소개 - 4줄 이상(background 제외) 작성
 - Result
 - 수행한 내용을 캡처 및 설명
 - 고찰
 - 과제를 수행하면서 느낀 점 작성
 - Reference
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능

Report Requirements

- Softcopy Upload

- 제출 파일
 - 보고서 + 소스파일 [하나의 압축 파일로 압축하여 제출(tar.xz)]
 - 보고서(.pdf. 파일 변환)
 - 소스코드(**Comment 반드시 포함**)
- Tar.xz 압축 및 해제 방법
 - **"Appendix C 참조"**
- 보고서 및 압축 파일 명 양식
- **OS_Assignment1 _수강분류코드_학번_이름** 으로 작성

수강요일	이론1 월6수5	이론2 목3	실습 금56
수강분류코드	A	B	C

- 예시 #1)-**이론(월6수5)만** 수강하는 학생인 경우
 - 보고서 OS_Assignment1_**A**_2024123456_홍길동.pdf
 - 압축 파일 명: OS_Assignment1_**A**_2024123456_홍길동.tar.xz
- 예시 #2)-**이론(월6수5 or 목3)과 실습** 모두 수강하는 학생인 경우
 - 보고서 OS_Assignment1_**C**_2024123456_홍길동.pdf
 - 압축 파일 명: OS_Assignment1_**C**_2024123456_홍길동.tar.xz
 - + **"해당 이론반 txt 파일 제출"**

Report Requirements

- 실습 수업을 수강하는 학생인 경우

- 실습 과목에 과제를 제출(.tar.xz)
- 이론 과목에 간단한 .txt 파일로 제출

📄 실습수업때제출했습니다.

2022-08-29 오후 3:58

텍스트 문서

0KB

- 이론 과목에 .txt 파일 미 제출 시 감점
- .tar.xz 파일로 제출 하지 않을 시 감점

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 9월 19일 목요일 23:59까지 제출
- 딜레이 받지 않음
 - 제출 마감 시간 내 미제출시 해당 과제 0점 처리(예외 없음)

Appendix

System Software Laboratory
School of Computer and Information Engineering
Kwangwoon Univ.

Appendix A. printk()

- **int printk(const char *fmt, ...);**
 - 커널에서 메시지를 출력하는 함수
 - 로그레벨 (declared in <linux/kernel.h>)
 - 로그레벨에 대한 문자열을 상수로 선언해 놓음

상수	문자열	의미
KERN_EMERG	"<0>"	System is unusable
KERN_ALERT	"<1>"	Action must be taken immediately
KERN_CRIT	"<2>"	Critical conditions
KERN_ERR	"<3>"	Error conditions
KERN_WARNING	"<4>"	Warning conditions
KERN_NOTICE	"<5>"	Normal but significant condition
KERN_INFO	"<6>"	Informational
KERN_DEBUG	"<7>"	Debug-level messages

- e.g. **printk**(KERN_INFO "just info\n")
- 사용법은 printf와 유사
- printk()로 출력된 메시지는 dmesg 명령어를 통해 확인

Appendix B. dmesg

▪ dmesg

- Linux command to print or control the kernel ring buffer
- printk()로 출력한 내용을 본 명령어를 통해 확인할 수 있음

▪ Usage

- 앞의 5개의 메시지만 보고 싶을 때,
 - **\$ dmesg | head -n 5**

```
sslslab@sslslab-VirtualBox:~$ dmesg | head -n 5
[ 0.000000] Initializing cgroup subsys cpuset
[ 0.000000] Initializing cgroup subsys cpu
[ 0.000000] Linux version 3.2.28-OSLAB-ASSISTANT (root@sslslab-VirtualBox) (gcc version
4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5) ) #3 SMP Thu Sep 3 00:56:08 KST 2015
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
sslslab@sslslab-VirtualBox:~$
```

- 뒤의 5개의 메시지만 보고 싶을 때,
 - **\$ dmesg | tail -n 5**

```
sslslab@sslslab-VirtualBox:~$ dmesg | tail -n 5
[ 3038.105356] [drm] Supports vblank timestamp caching Rev 1 (10.10.2010).
[ 3038.105357] [drm] No driver support for vblank timestamp query.
[ 3038.105359] [drm] Initialized vboxvideo 1.0.0 20090303 for 0000:00:02.0 on minor 0
[ 3040.072008] init: plymouth-stop pre-start process (1400) terminated with status 1
[ 3042.596238] eth0: no IPv6 routers present
sslslab@sslslab-VirtualBox:~$
```

Appendix B. dmesg(Cont'd)

- Usage
 - 특정 문자가 포함된 메시지 열을 보고 싶을 때
 - \$ dmesg | grep XXX**

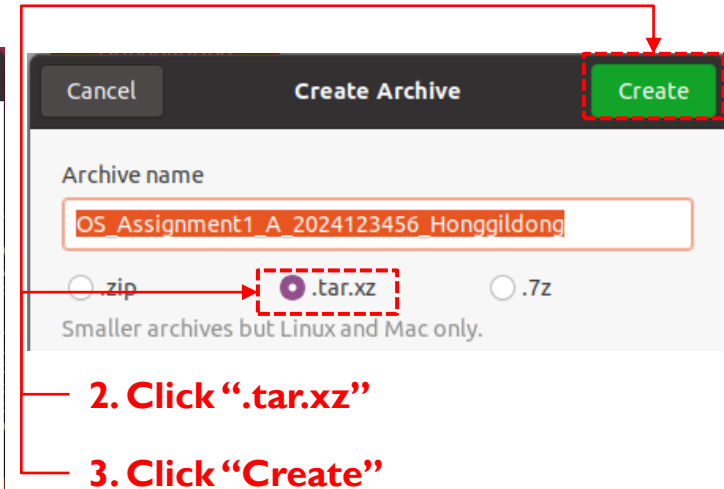
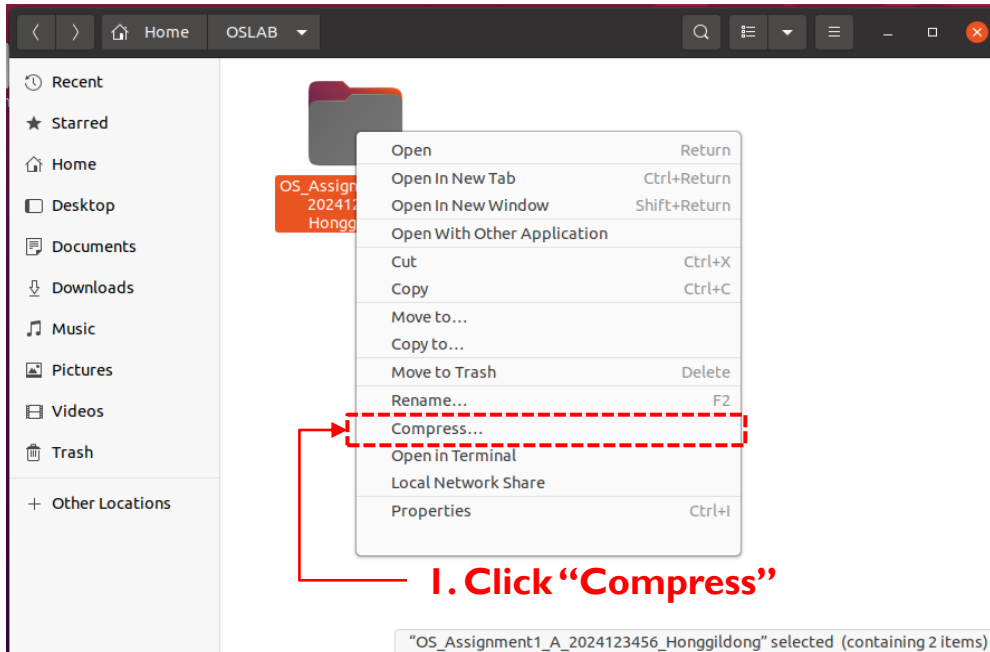
```
sslab@sslab:~$ dmesg |grep SHY  
[    0.000000] OS_lecture_SHY
```

- 특정 문자가 포함된 메시지 열이 몇 번째 열인지 보고 싶을 때
 - \$ dmesg | grep XXX -n**

```
sslab@sslab:~$ dmesg |grep SHY -n  
388:[    0.000000] OS_lecture_SHY
```

Appendix C. tar.xz compression

■ Solution 01. Compress using GUI



Appendix C. tar.xz compression(Cont'd)

- Solution 02. Compress using CLI

```
test12@ubuntu:~/OSLAB$ ls
OS_Assignment1_A_2024123456_Honggildong
test12@ubuntu:~/OSLAB$ tar Jcvf OS_Assignment1_A_2024123456_Honggildong.tar.xz OS_Assignment1_A_2024123456_Honggildong
OS_Assignment1_A_2024123456_Honggildong/
OS_Assignment1_A_2024123456_Honggildong/code.c
OS_Assignment1_A_2024123456_Honggildong/Report.pdf
```

- 압축방법(Compress)

- \$tar -Jcvf [File_name.tar.xz] [Target]
- \$tar -Jcvf [압축파일이름.tar.xz] [압축대상]

- 압축해제(Extract)

- \$tar -Jxvf [File_name.tar.xz]
- \$tar -Jxvf [압축된 파일 이름.tar.xz]