

2023년 2학기 운영체제 & 운영체제 실습

Assignment 1

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

Requirements

- Ubuntu 16.04.5 Desktop 64bits 환경에서 채점
- Copy 발견 시 0점 처리
- 보고서
 - 표지
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
 - 과제 이름 → assignment 1
 - 아래의 내용은 보고서에 필히 포함
 - **Introduction** : 4줄 이상(background 제외) 작성
 - **Result** : 수행한 내용을 캡처 및 설명
 - **고찰** : 과제를 수행하면서 느낀점 작성
 - **Reference**
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능
 - **Source**
 - [Assignment 1-3] 에서 수정한 코드 첨부

Requirements (cont'd)

- **Softcopy만 작성(Hardcopy 받지 않음)**
- **제출 파일**
 - 보고서(.pdf) + Source file 하나의 압축파일로 압축하여 제출(**tar.gz**)
 - Tar 압축 및 해제 방법
 - 압축 시 → tar -zcvf [압축 파일명].tar.gz[폴더 명]
 - 해제 시 → tar -zxvf 파일명.tar.gz
- **보고서 및 압축 파일 명 양식**
 - **os_과제번호_학번_수강분류코드**

수강요일	이론_화목	이론_금12	실습
수강분류코드	A	B	C
 - e.g
 - 이론_월수 수강하는 학생인 경우
(보고서) os_1_20221234567_A.pdf (압축파일) os_1_20221234567_A.tar.gz
 - 이론_금12 수강하는 학생인 경우
(보고서) os_1_20221234567_B.pdf (압축파일) os_1_20221234567_B.tar.gz
 - 실습 수강하는 학생인 경우
(보고서) os_1_20221234567_C.pdf (압축파일) os_1_20221234567_C.tar.gz

Requirements (cont'd)

- **실습 수업을 수강하는 학생인 경우**
 - 실습 과목에 과제를 제출(.tar.gz)
 - 이론 과목에 간단한 .txt 파일로 제출

 실습수업때제출했습니다.

2022-08-29 오후 3:58 텍스트 문서

0KB

- **이론 과목에 .txt 파일 미 제출 시 감점**

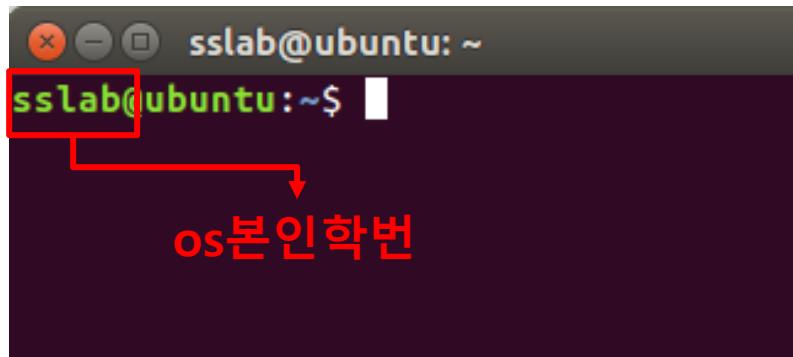
- **과제 제출**

- **KLAS – 강의 과제 제출**
- **2023년 9월 21일 목요일 23:59까지 제출**
- **딜레이 받지 않음**
 - 제출 마감 시간 내 미제출시 해당 과제 0점 처리
 - 교내 서버 문제 발생 시, 메일로 과제 제출 허용

Assignment 1-1

▪ Linux Installation

- 과제 요구 사항
 - 계정 ID “os학번”으로 할 것
 - (e.g. os2022123456)
 - 터미널 화면 캡쳐 후 보고서에 첨부



```
sslab@ubuntu:~$ os본인학번
```

▪ Linux command

- 과제 요구 사항
 - 각 해당하는 리눅스 명령어 사용하여 캡쳐 후 보고서에 첨부
 - 1. “assignment1”명의 디렉터리 생성
 - 2. assignmnet1 디렉터리 이동 후 “os.txt”명의 빈 파일 생성
 - 3. os.txt를 os_copy.txt명으로 복사
 - 4. os_copy.txt의 권한을 모든 대상에게 읽기만 부여
 - 5. os.txt에 os_본인학번 작성 후 터미널에 출력

```
sslab@ubuntu:~$ ls  
sslab@ubuntu:~$ cd assignmnet1  
①  
②  
sslab@ubuntu:~/assignmnet1$ ls  
os.txt  
sslab@ubuntu:~/assignmnet1$ ls -al  
③  
total 8  
drwxrwxr-x 2 sslab sslab 4096 Aug 29 00:25 .  
drwxr-xr-x 17 sslab sslab 4096 Aug 29 00:24 ..  
-rw-rw-r-- 1 sslab sslab 0 Aug 29 00:25 os_copy.txt  
-rw-rw-r-- 1 sslab sslab 0 Aug 29 00:25 os.txt  
sslab@ubuntu:~/assignmnet1$ cp os.txt os_copy.txt  
④  
sslab@ubuntu:~/assignmnet1$ chmod 644 os_copy.txt  
sslab@ubuntu:~/assignmnet1$ echo os_2022123456 > os.txt  
⑤  
sslab@ubuntu:~/assignmnet1$ cat os.txt  
os_2022123456  
sslab@ubuntu:~/assignmnet1$
```

Assignment 1-2

▪ Kernel 4.19.67 Compile

- 우분투 내에서 Kernel을 다운로드 하고, 컴파일의 모든 과정을 terminal과 vi를 이용하여 보고서에 첨부

▪ 과제 요구 사항

- 커널 컴파일 과정을 terminal과 vi를 사용하여 캡쳐
- 각 명령어가 어떠한 기능을 하였는지 간단히 서술
- 4.19.67 커널로 재 부팅 후 버전 확인
- "uname -r" 의 결과가 "**4.19.67-본인학번**" 이 나오도록 진행할 것

```
sslab@ubuntu:~$ uname -r  
4.19.67-본인학번  
sslab@ubuntu:~$ █
```

Assignment 1-3

▪ Linux agp...이 실행되는 지점에서 Linux Kernel Message 출력

- Linux agp...이 실행되는 지점에 커널 메시지가 다음과 같이 출력되도록 커널 코드 수정 -----(1)
- Linux agp... 을 실행시키는 함수의 함수명과 argument의 값을 출력하도록 커널 코드 수정 -----(2)
- **Linux Kernel Message 양식**

os본인학번_Linux agpgart interface v0.103	(1)
os본인학번_arg in <i>function_name</i> (<i>argument</i>)	(2)

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



argument

▪ Hints

- cscope, ctags 등을 활용

Assignment 1-3

- Linux agp...이 실행되는 지점에서 Linux Kernel Message 출력
 - 과제 요구사항
 - printk() 함수 사용
 - 로그 레벨은 KERN_INFO 사용 (강의 자료 "[Appendix A](#)" 참고)
 - 출력 메시지는 7 page의 [[Linux Kernel Message 양식](#)] 참고.
 - dmesg , grep으로 확인 시 (1), (2) 메시지가 연달아 출력되어야 한다
 - 강의 자료 "[Appendix B](#)" 참고
 - 주의사항 1, 2를 참고 (9 page)
 - 보고서에 다음의 내용을 필히 포함
 - 리눅스 커널 코드에서 수정한 부분을 명시
 - 소스코드 path 작성
 - 검색한 캡쳐 화면 첨부
 - 결과 화면 캡쳐
 - Softcopy 제출 시
 - 수정한 커널 코드 파일을 제출
 - ex) init/main.c를 수정 시 main.c 파일을 첨부하여 제출
 - 수정한 커널 코드 파일 명 수정 하지 않음

Assignment 1-3

- 주의 사항 1 (cont'd)
 - 프린트 되는 문자열은 중복되어 여러 번 나오면 안됩니다.

[ex] 잘못된 예시

1010: [4.107586] os2019110613_Linux agpgart interface v0.103

1011: [4.107590] os2019110613_arg in *function_name* (*argument*)

...

1404: [5.107586] os2019110613_Linux agpgart interface v0.103

1405: [5.107586] os2019110613_arg in *function_name* (*argument*)

- 아래와 같이 중복 없이 한 set로 나와야 인정합니다.
- 두 문구 중 어느 하나라도 중복으로 출력되면 감점입니다.

[ex] 올바른 예시

1404: [5.107586] os2019110613_Linux agpgart interface v0.103

1404: [5.107586] os2019110613_arg in *function_name* (*argument*)

▪ 주의 사항 2 (cont'd)

- 제대로 된 위치에서 문자열을 출력해야 인정합니다.

- 본인이 기술한 "start_kernel()"에서 Linux agp...이 실행되는 지점"에 기반하여 수정한 코드만 인정

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**

```
sslab@ubuntu:~$ dmesg | head -n 5
[ 0.000000] SSLAB in start_kernel()
[ 0.000000] Linux version 4.19.67-SSLAB (sslab@ubuntu) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.19.67-SSLAB root=UUID=71e14006-ab36-4801-a116
[ 0.000000] KERNEL supported cpus:
[ 0.000000]   Intel GenuineIntel
sslab@ubuntu:~$ █
```

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

```
sslab@ubuntu:~$ dmesg | tail -n 5
[ 15.397111] Bluetooth: RFCOMM ver 1.11
[ 33.377119] atkbd serio0: Unknown key pressed (translated set 2, code 0x72 on isa0060/serio0).
[ 33.377124] atkbd serio0: Use 'setkeycodes 72 <keycode>' to make it known.
[ 33.377460] atkbd serio0: Unknown key released (translated set 2, code 0x72 on isa0060/serio0).
[ 33.377463] atkbd serio0: Use 'setkeycodes 72 <keycode>' to make it known.
sslab@ubuntu:~$ █
```

Appendix B. dmesg

- 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
```