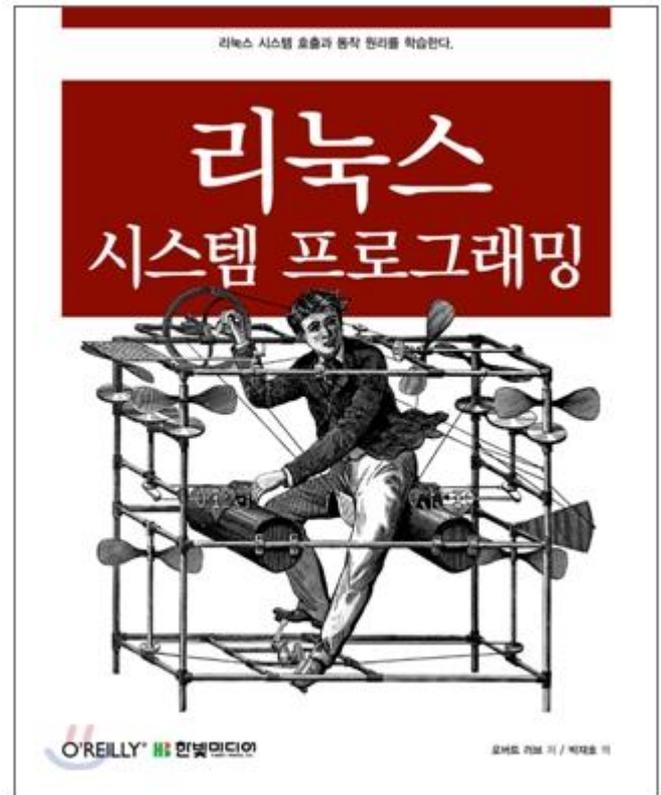


유닉스 프로그래밍 및 실습

본 강의에 들어가기 전

교재 소개

- ▶ 리눅스 시스템 프로그래밍
 - ▶ 원제: Linux System Programming
 - ▶ 로버트 러브 저, 박재호 역
 - ▶ O'REILLY 한빛미디어
- ▶ 참고사이트
 - ▶ <http://linux.die.net/man/>
 - ▶ <http://seeseepbworks.com/f/linux.pdf>



교재 목차 (1)

- ▶ 1장 소개와 핵심 개념
 - ▶ 시스템 프로그래밍
 - ▶ API와 ABI
 - ▶ 표준
 - ▶ 리눅스 프로그래밍 개념
 - ▶ 시스템 프로그래밍을 시작하며
- ▶ 2장 파일 입출력
 - ▶ 파일 열기
 - ▶ read()로 읽기
 - ▶ write()로 쓰기
 - ▶ 동기식 입출력
 - ▶ 직접 입출력
 - ▶ 파일 닫기
 - ▶ lseek()으로 탐색하기
 - ▶ 위치를 지정한 읽기와 쓰기
 - ▶ 파일 잘라내기
 - ▶ 다중 입출력
 - ▶ 커널 내부
 - ▶ 결론
- ▶ 3장 버퍼 입출력
 - ▶ 사용자 버퍼 입출력
 - ▶ 표준 입출력
 - ▶ 파일 열기
 - ▶ 파일 기술자로 스트림 열기
 - ▶ 스트림 닫기
 - ▶ 스트림에서 읽기
 - ▶ 스트림에 쓰기
 - ▶ 버퍼 입출력을 활용하는 예제 프로그램
 - ▶ 스트림 탐색하기
 - ▶ 스트림 강제 출력(버퍼 비우기)
 - ▶ 오류와 EOF
 - ▶ 관련된 파일 기술자 얻기
 - ▶ 버퍼 제어하기
 - ▶ 스레드 안전
 - ▶ 표준 입출력 비판
 - ▶ 결론

교재 목차 (2)

- ▶ 4장 고급 파일 입출력
 - ▶ 이합/집산 입출력
 - ▶ epoll(Event Poll) 인터페이스
 - ▶ 파일을 메모리에 사상하기
 - ▶ 일반 파일 입출력을 위한 조언
 - ▶ 동기화, 동기식, 비동기식 연산
 - ▶ 입출력 스케줄러와 성능
 - ▶ 결론
- ▶ 5장 프로세스 관리
 - ▶ 프로세스 ID
 - ▶ 새로운 프로세스 실행하기
 - ▶ 프로세스 종료하기
 - ▶ 자식 프로세스 종료 기다리기
 - ▶ 사용자와 그룹
 - ▶ 세션과 프로세스 그룹
 - ▶ 데몬
 - ▶ 결론
- ▶ 6장 고급 프로세스 관리
 - ▶ 프로세스 스케줄링
 - ▶ 프로세서 양보하기
 - ▶ 프로세스 우선순위
 - ▶ 프로세서 친화도
 - ▶ 실시간 시스템
 - ▶ 자원 제한
- ▶ 7장 파일과 디렉터리 관리
 - ▶ 파일과 메타자료
 - ▶ 디렉터리
 - ▶ 링크
 - ▶ 파일 복사와 이동
 - ▶ 디바이스 노드
 - ▶ 대역 외 통신
 - ▶ 파일 사건 감시하기

교재 목차 (3)

▶ 8장 메모리 관리

- ▶ 프로세스 주소 공간
- ▶ 동적 메모리 할당하기
- ▶ 자료 세그먼트 관리하기
- ▶ 익명 메모리 사상
- ▶ 고급 메모리 할당
- ▶ 메모리 할당 디버깅
- ▶ 스택 기반 할당
- ▶ 메모리 할당 메커니즘 선택하기
- ▶ 메모리 조작하기
- ▶ 메모리 잠그기
- ▶ 기회주의적인 할당 전략

▶ 9장 시그널

- ▶ 시그널 개념
- ▶ 기초적인 시그널 관리
- ▶ 시그널 보내기
- ▶ 재진입성
- ▶ 시그널 집합
- ▶ 시그널 차단하기
- ▶ 고급 시그널 관리
- ▶ 페이로드와 더불어 시그널 보내기
- ▶ 결론

▶ 10장 시간

- ▶ 시간을 표현하는 자료 구조
- ▶ POSIX 시계
- ▶ 현재 시각 얻기
- ▶ 현재 날짜와 시각 설정하기
- ▶ 시간 다루기
- ▶ 시스템 시계 조율
- ▶ 잠들기와 기다리기
- ▶ 타이머

▶ 부록 C 언어를 위한 GCC 확장

개발환경 구축

- ▶ Xshell4 설치

 - ▶ www.netsarang.co.kr

- ▶ lily.mmu.ac.kr 계정 신청

- ▶ USB등으로 개인 폴더 유지

개발 과정(1)

- ▶ Vi 사용법 익히기
 - ▶ UNIX 및 실습 4장 참조 (http://lily.mmu.ac.kr/lecture/12u1/UNIX_4.pdf)
- ▶ 기초 개발과정(1)
 - ▶ 에디터로 프로그램 작성
 - ▶ vi abc.c

```
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
}
```
 - ▶ 컴파일
 - ▶ gcc abc.c
 - ▶ 실행
 - ▶ ./a.out

개발 과정(2)

▶ 기초 개발과정(2)

▶ 컴파일시 실행파일 이름 주기 (-o)

- ▶ gcc -o abc abc.c

- ▶ ./abc

▶ 컴파일만 하기 (-c)

- ▶ gcc -c abc.c

- ▶ ls

▶ make 사용법

- ▶ http://lily.mmu.ac.kr/lecture/04pl1/pre_03.htm