과제1 안내

컴퓨터공학과 장대희

Linux 크래시 실습

- 개요
  - 시스템콜 등 커널에서 실행될 코드를 분석하고 크래시를 유발하는 코드 삽입
  - 수정된 커널소스를 새로 컴파일하여 QEMU 에뮬레이터로 부팅
  - 부팅중 또는 부팅후 특정 명령 수행시 삽입된 크래시 유발코드가 실행

- 크래시 유발코드 (C 언어)

  long int* boom = (long int*)0xcafebabe;

  *boom = (본인의 학번을 16진수로 변환 후 0x00FFFFFF 와 AND 한 값);

과제 진행절차

- 아래 순서로 진행
  - ✓ 1. Ubuntu 환경준비 (20.04, 22.04 권장)
  - ✓ 2. 리눅스 커널소스 다운로드 https://www.kernel.org/
    - ✓ 커널버전: 6.2.8
  - ✓ 3. QEMU-ARM 에뮬레이터 설치:
    - ✓ sudo apt install qemu-system-arm
  - ✓ 4. ARM 크로스컴파일러 및 개발보조도구 설치:
    - ✓ sudo apt install gcc-arm-linux-gnueabi make flex bison
  - ✓ 5. ARM 리눅스 커널컴파일 (아래 링크 등 참고)
    - ✓ https://jasonblog.github.io/note/arm_emulation/compiling_linux_kernel_for_qemu__arm_emulator.html
    - ✓ https://github.com/surajx/qemu-arm-linux
    - ✓ QEMU 지원 config 로 (versatilepb 등) 컴파일 해야함 유의.
    - ✓ 최신커널은 DTB파일 추가해야함 유의.
    - ✓ https://community.arm.com/oss-platforms/w/docs/525/device-tree

과제 진행절차

- 아래 순서로 진행
  - ✓ 6. BusyBox 소스코드 다운후 ARM 크로스컴파일 (아래 링크 등 참고)
  - ✓ https://jasonblog.github.io/note/arm_emulation/busybox_for_arm_on_qemu.html
  - ✓ https://github.com/surajx/qemu-arm-linux
    - ✓ static 컴파일 해야함 유의.
  - ✓ 7. 에뮬레이터 (qemu-system-arm) 을 이용해서 BusyBox 와 함께 수정 전 OS 구동 후 쉘 획득
    - ✓ 1단계 완성
  - ✓ 8. ARM 리눅스 커널 소스코드 분석 후 커널소스코드 수정.
    - ✓ 크래시 유발코드를 삽입할 부분 찾기
  - ✓ 9. 수정된 ARM 리눅스 커널을 QEMU 로 부팅시킨 후 커널 크래시 관찰
    - ✓ 2단계 최종 완성

제출물

❖다음 파일들을 수업조교에게 email 로 제출 (yuri0329@khu.ac.kr)

- 1단계, 2단계 각각 완료화면 스크린샷
- zImage.back (커널을 수정하기 전 쉘 획득까지 정상실행되는 커널이미지)
- zImage (커널을 수정하여 컴파일한 커널이미지)
- 수정한 커널소스 파일
- rootfs.img.gz (gzip된 BusyBox 파일시스템)
- dtb 파일 (Device Tree Blob)
- start.sh (QEMU 구동 스크립트)
- 1단계/2단계 가 정상적으로 재현/검증 되어야 제출이 인정됨.

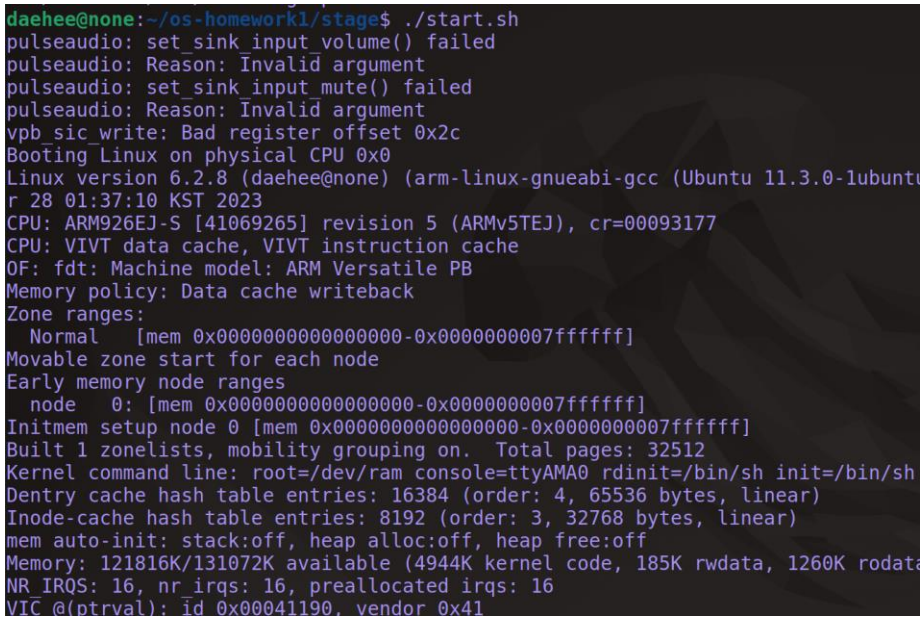
❖아래 스크린샷 참고

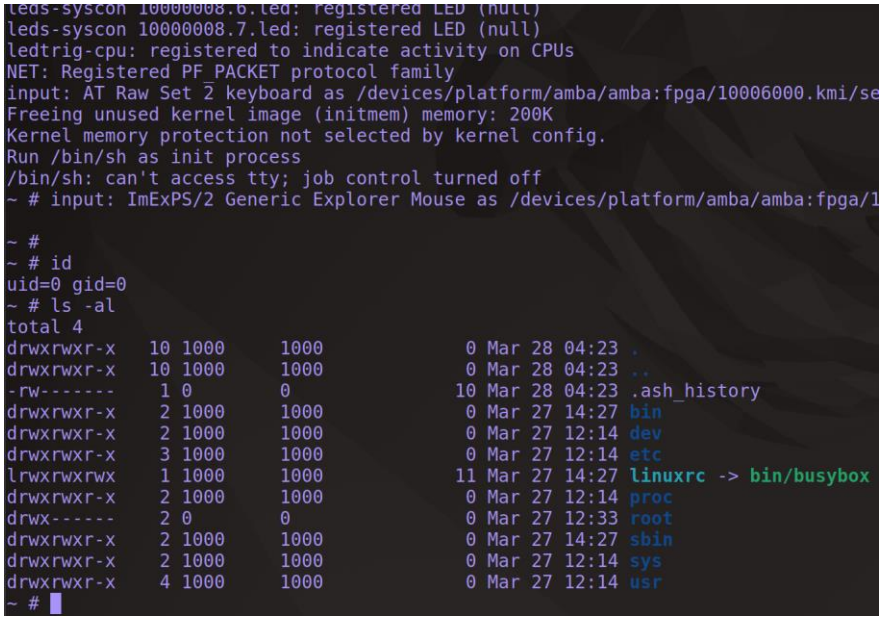
```
daehee@none:~$ ls
a  a.c  a-dynamic  a-static  lab  os-homework1  snap
daehee@none:~$ cd os-homework1/
daehee@none:~/os-homework1$ ls
busybox-1.36.0          linux-6.2.8        linux-6.2.8.tar
busybox-1.36.0.tar.bz2  linux-6.2.8.patch  stage
daehee@none:~/os-homework1$ cd stage
daehee@none:~/os-homework1/stage$ ls
rootfs.img  rootfs.img.gz  start.sh  versatile-pb.dtb  zImage  zImage.back
daehee@none:~/os-homework1/stage$ cat start.sh
#!/bin/sh
qemu-system-arm -M versatilepb -m 128M -kernel zImage -dtb versatile-pb.dtb -initrd rootfs
.img.gz -append "root=/dev/ram console=ttyAMA0 rdinit=/bin/sh init=/bin/sh" -nographic
daehee@none:~/os-homework1/stage$
```

결과화면

❖과제 1단계 정상 수행시 아래와 유사한 화면을 확인해야함

- QEMU 로 정상 부팅후 BusyBox 마운트한뒤 쉘 획득

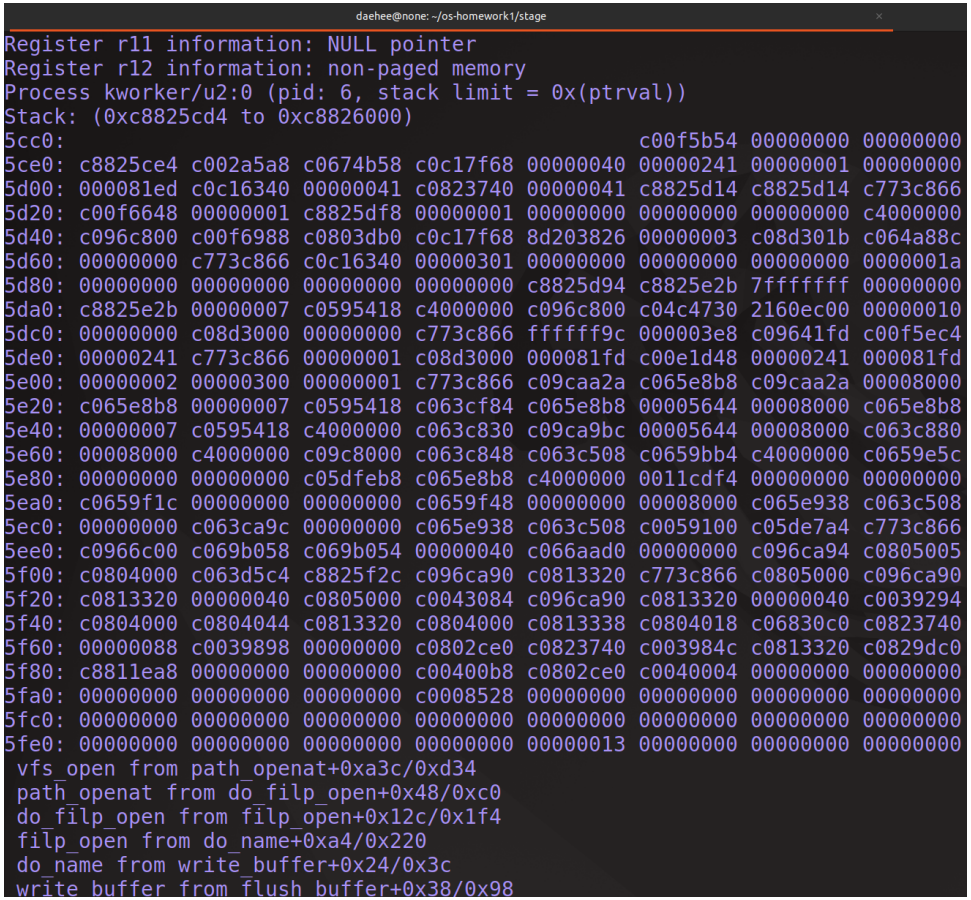




결과화면

❖과제 2단계 정상 수행시 아래와 유사한 화면을 확인해야함

- 메모리 접근오류로 커널패닉 확인



제출기한 및 평가

❖제출기한

- 4월 4일 수업시간 전까지 (1차)
  - ✓ 수업시간때 과제 수행방법 추가안내
- 4월 11일 수업시간 전까지 (2차)
  - ✓ 수업시간때 과제 수행방법 추가안내 (그대로 따라하면 되는 수준)
- 4월 18일 수업시간 전까지 (3차)

❖점수: 5점만점 (5% 성적반영)

- 과제1 5%, 과제2 5%. 총합 10% 성적비율

❖평가방식

- 최초 제출자 1인-> 가산점 1점
  - ✓ 중간고사+기말고사 점수가 만점이 아닌경우 +1점 (만점이면 가산점 없음)
- 1차 평가
  - ✓ 1차 기한이내 제출완료 (코드검증 완료) 시 5점
- 2차 평가
  - ✓ 2차 기한이내 제출완료 (코드검증 완료) 시 4점
- 3차 평가
  - ✓ 3차 기한이내 제출완료 (코드검증 완료) 시 3점
  - ✓ 3차 기한이내 제출완료 (코드검증 실패) 시 1점


Q
A