

GDB

# 디버거 교육 목적

1. 시스템 콜 코드 분석하기 위해
2. 리눅스 프로세스 관리에서 스케줄러 분석에 활용
3. 리눅스 디바이스 드라이버 수업에서 활용
4. 향후 개발에 사용

# 툴체인(Toolchains)

- GNU toolchain = GCC + binutils + C library + GDB
- GCC
  - GNU Compiler Collection - C, C++
- Binutils
  - assembler, linker and utilities
  - nm, objdump, readelf, gdb
- GDB
  - the GNU debugger

# GDB

- 서버와 임베디드 장비는 PC 운영체제처럼 GUI가 없음
- 상황에 따라 gdb를 통해 디버깅 + 개발

# GDB commands

- Run `help <command-name>`
- 단축 커맨드
  - `c` = `co` = `cont` = `continue`
  - `s` = `step` and `si` = `stepi`

# GDB commands

- `gdb <program>`
- `gdb -p <pid>`
- `(gdb) run`

# step

- step
  - 한 라인 수행 만약 함수를 만나면 함수 안으로 진입
- next
  - 한 라인 수행 함수를 만나도 다음 라인 진행
- stepi and nexti
  - 어셈블리어 단위로 수행

# 실행

- continue
  - 다음 브레이크 포인트를 만날 때까지 수행
- finish
  - 현재 함수가 반환 할 때까지 수행



# Breakpoints

- break <location>
  - 특정 위치에서 멈춘다.
  - Location
    - memory addresses ("\*0x7c00")
    - main.c:50
- delete, disable, enable.
  - 브레이크 포인트 수정

# Watchpoints

- watch <expression>

# 출력

- x prints
  - 메모리 값 출력
  - x/x: hexadecimal
  - x/i: assembly
- print evaluates
- p \*((struct 데이터 타입 \*) 0x20000

# 출력

- info registers
  - CPU 레지스터 출력
- info frame
  - 현재 스택 프레임 출력
- list <location>
  - 소스 코드 출력
- backtrace
  - 함수 콜 스택 출력
- info threads
  - Thread 리스트 출력
- info breakpoints
  - Breakpoint 리스트 출력

# fork debugging

- set follow-fork-mode child
  - 기본 값은 parent
- show follow-fork-mode

# GDB frontends

- TUI
- CGDB
- DDD
- VSCODE

# vscode를 이용한 디버깅

- <https://code.visualstudio.com/docs/cpp/config-linux>
- tasks.json
  - compiler build settings
- launch.json
  - debugger settings
- c\_cpp\_properties.json
  - compiler path and IntelliSense settings
- 디버그 콘솔
  - GDB 명령 가능
  - 예) -exec set follow-fork-mode child

# gdb를 이용한 디버깅 실습(10분)

- GDB를 이용해 아래 4가지 프로그램을 디버깅한다.
- **copy**
  - tlp-dist/fileio/copy
- **fork**
  - tlp-dist/procexec/fork\_whos\_on\_first
- **exec**
  - tlp-dist/procexec/f\_exec
- **pipe**
  - tlp-dist/pipes/simple\_pipe



# vscode를 이용한 디버깅 실습(15분)

- VSCODE를 활용하여 아래 4가지 프로그램을 디버깅한다.
- **copy**
  - `tlpi-dist/fileio/copy`
- **fork**
  - `tlpi-dist/procexec/fork_whos_on_first`
- **exec**
  - `tlpi-dist/procexec/f_exec`
- **pipe**
  - `tlpi-dist/pipes/simple_pipe`

# TOY 프로젝트 실습 과제(20분)

- TOY 프로젝트용 launch.json을 작성 과제
  - TOY 프로젝트를 vscode를 이용해서 디버깅 실습
  - main 함수에 브레이크 포인트