

이름 gdb-the gnu debugger

개요 `gdb [-help] [-nh] [-nx] [-q] [-batch] [-cd=dir] [-f] [-b bps]`

`[-tty=dev] [-s symfile] [-e prog] [-se prog] [-c core] [-p proclD]`

`[-x cmds] [-d dir] [prog|prog proclD|prog core]`

설명: 디버거의 목적은 다른 프로그램이 실행 중에 그 안에서 무슨 일이 일어나는지 확인하게 해 주고 문제가 생겼을 때 무슨 일을 하는 중이었는지 알려준다.

Gdb는 4가지 주요 작업을 할 수 있다.

1. 프로그램을 실행시킨다. 작업에 영향을 줄만 한 일은 뭐든지 명시한다.
2. 특정한 조건에서 프로그램을 중지시킨다.
3. 프로그램이 중단된 경우 무슨 일이 일어났는지 확인한다.
4. 버그의 문제를 고치고 다른 내용을 더 알아보려 갈 수 있게 프로그램을 수정한다.

Gdb를 c, c++, fortran, modula-2를 디버그하는데 쓸 수 있다. Gdb는 gdb 커맨드로 실행되고 실행 되면 터미널의 커맨드를 quit 커맨드로 중단시킬 때 까지 읽는다. help 커맨드로 gdb에 대한 온라인 도움을 받을 수도 있다.

Gdb는 아무런 옵션이나 명령어 없이 실행할 수 있지만, 보통 1,2개의 명령을 추가로 써서 실행할 프로그램을 명시한다. 예) `gdb program`

추가로 프로그램에 주요 파일을 명시할 수 있다. 예) `gdb program core`

실행중인 프로세스의 디버그를 원할 경우 프로세스 id를 정확히 쓸 수도 있다.

예) `gdb program 1234`, `gdb -p 1234` (1234라는 파일 이름이 있을 경우 주의), -p는 파일이름을 제외한다.

주요 gdb 명령어들:

`break [file:]function` : file의 중단지점을 function으로 설정한다,

`run [arglist]` : arglist를 가진 프로그램 실행,

`bt` : 프로그램 스택을 역추적한다, `print expr` : expr을 출력함, `c` : 프로그램 계속 실행(정지나 중지 후)

`next` : 다음 프로그램 라인 실행(정지후), `edit [file]function` : 현재 정지한 프로그램 라인 확인,

`list [file]function` : 현재 중지된 프로그램의 텍스트 출력, `step` : 다음 프로그램 라인 실행(정지후) 함수 첫라인으로 이동, `help [name]` : name 커맨드의 내용 확인, `quit` : gdb 종료