

<<< 유용한 기본 유틸리티입니다. 실행해 보세요.. -----

- 01) alias (숨겨진 명령을 볼수있습니다.) ll, mv, cp, etc 확인..
- 02) xxd filename (바이너리 값을 확인할 수 있습니다.) 또는 od
- 03) ddd (그래픽 모드에서 실행하여 gdb를 사용할 수 있습니다.)
- 04) gcc -W -Wall -v --save -o test test.c
(--save 중간단계를 모두 저장해 줍니다.)
- 05) size filename (섹션별 크기를 보여줌)
- 06) readelf -a 실행파일 (실행파일에 있는 elf 파일 내용 보기)
: 코어파일 분석 시 사용)
- 07) objdump -S 실행파일 (심벌 옵션 보기)
(-d 는 순수 어셈만 보임)
- 08) 컴파일시 -O0, -O1, -O2, -O3 를 각각 주어 생성 후
- 09) time ./실행파일 (각 단계별 실행시 소요 시간을 보여줍니다)
(최적화 후 성능비교)
- 10) strip -R .comment -s 실행파일
(.note 섹션을 제거 파일 크기를 줄여줌)
- 11) shift + pgup, shift + pgdn
(터미널 창에서 지나간 화면 보기)

<<< vi에서 실행해 보세요.. -----

- 01) C언어 소스코드에서 v 하고 영역 설정 후 =키를 타이핑하시면
탭이 정렬됩니다.

<<< binutils 계열 프로그램들 -----

- 01) objdump, 02) addr2line, 03) ar, 04) gprof, 05) ld, 06) nm,
07) objcopy, 08) objdump, 09) ranlib, 10) readelf, 11) size,
12) strip

<<< gdb 상에서 실행해 보세요 -----

- 01) p \$ebx (레지스터의 값 보기) \$eip, \$eax, \$ecx, etc
- 02) h 또는 help (명령어 목록을 볼 수 있습니다.)
p, b, r, n, etc
- 03) info locals (지역변수들과 값을 알수 있습니다.)
- 04) info frame (스택영역의 정보를 볼 수 있습니다.)
- 05) info variables (전역변수를 열람할 수 있습니다.)
- 06) bt 또는 backtrace (콜스택과 같이 역추적하여
함수 호출관계를 볼 수 있습니다.)
- 07) x/10i main (main 함수의 시작부터 어셈블리로 보여줌)
i,s,f,a,b,h,w,g etc
- 08) disas main
- 09) r 하고 signal SIGKILL (디버그 모드에서 실행중에
시그널을 보낼 수 있습니다.)
- 10) b, cl, d, n, s (브레이크 포인트 지정, 제거, 모두 제거,
한 스텝씩 안으로, 밖으로)
- 11) watch (변수의 값이 바뀔때 마다 정지(브레이크))
- 12) display 변수명 (변수의 값을 매번 표시) undisplay
- 13) call 함수명(입력값) (함수별로 별도로 실행해 볼 수 있음)
(실행 후 리턴값 출력됨)
- 14) 이상의 내용을 “ gdb 실행파일 코어파일 ” 순으로
실행하여 적용해보기
- 15) 코어 파일 생성이 안되는 경우
셸에서) ulimit -c unlimited