REPORT

|  |  |
| --- | --- |
| 담당교수 | 장경식 |
| 과목명 | 시스템프로그래밍 |
| 이름 | 이아름 |
| 학부 | 컴퓨터공학부 |
| 학번 | 2013136089 |

시스템 프로그래밍

Lab5 – make 유틸리티 사용

1. **실습 과제**

**Lab-5 : make 유틸리티 사용**

1. 샘플 소스 파일 mtest.c foo.c boo.c bar.c를 만든다.

- foo.c : InFoo() 함수를 정의한다. 함수의 내용은 자유.

- boo.c : InBoo() 함수를 정의한다. 함수내용은 자유.

- bar.c : InBar() 함수를 정의한다. 함수내용은 자유.

- mtest.c : InFoo(), InBoo(), InBar() 함수를 호출한다.

1. mtest.c를 컴파일하여, mtest 라는 실행파일을 생성하는 makefile을 작성한다.
2. make 유틸리티를 사용하여 mtest를 생성하고, 쉘에서 실행한다.
3. 컴파일시 –g 옵션을 추가하여 디버그 정보를 생성한다.
4. gdb를 사용하여 mtest 파일에서 Infoo, InBoo, InBar 함수를 호출하는 과정을 trace 한다.

\* Due day : 10.21(금)

1. **실습 내용**

코드 작성

|  |
| --- |
| * + - * 1. bar.c |
| * + - * 1. boo.c |
| * + - * 1. foo.c |
| * + - * 1. mtest.c |

컴파일 후 mtest라는 실행파일을 생성하는 makefile 작성

|  |
| --- |
| ★ Makefile의 필요성  프로그램을 개발하게 되면 보통 라인 수가 증가하게 되어 하나의 파일만으로 프로그램을 구성하는 것은 어려운 일이다. 그래서 보통은 각각의 기능을 작은 모듈로 나누어 여러 파일로 나누어 개발을 하게 된다. 이 파일들은 서로 관계를 가지고 있는데, 어느 하나를 필요에 의해 바꾸게 되었을 때 그 파일에 있는 모듈(함수)를 이용하는 다른 파일도 새롭게 컴파일 되어야 한다. 하지만 파일 수가 많은 경우 이를 일일이 컴파일을 하게 될 때, 그 불편함과 함께 컴파일 하지 않아도 될 것도 컴파일을 하게 될 수 있고, 컴파일 해야 할 것도 미쳐 못하게 되어 링크시 에러를 발생하게 되는 경우도 있게 된다.  이런 상황에서 자동적으로 관계 있는 것만 새롭게 컴파일 하거나, 여러 개의 입력 파일로부터 출력 파일을 생성할 때 make 유틸리티는 필요하게 된다. 따라서, 컴파일 시간을 절약할 수 있고, 파일 수가 많을 시에 빌드와 관련된 파일을 한 곳에 모을 수 있다.    생성된 mtest    실행 |

Gdb 디버깅 모드로 진입

|  |
| --- |
| help 입력 시 나오는 명령어 설명  Gdb로 실행    gdb를 사용하여 mtest 파일에서 Infoo, InBoo, InBar 함수를 호출하는 과정을 trace  세 함수에 b명령어를 이용하여 break points 설정      명령어 run을 이용하여 실행    Continue 명령어를 이용하여 함수 호출 과정 확인 |

1. **고찰**

이번 과제에서는 makefile을 이용하여 더욱 편하게 여러 파일을 컴파일 하는 법과, gdb를 이용하여 디버깅 하는 법을 익힐 수 있었다. 여러 개의 파일을 한꺼번에 관리하다 보면 분명히 오늘 배운 makefile이 이롭게 쓰일 것이라고 생각된다. 순서대로 차근차근 하지 않아서 중간에 꼬여서 과제를 하는 데 시간이 조금 걸렸던 것 같다. Gdb로 디버깅 하는 것 또한 매우 흥미로웠다. GUI가 제공되는 비주얼스튜디오에서 화살표를 눌러가며 디버깅했던 게 내가 가지고 있는 디버깅의 기억의 전부인데, 이렇게 프롬프트 화면에서 직접 중단점을 설정해가며 디버깅을 하는 것이 흥미로웠다. Gdb에 대해 알아보며 명령어들을 훑어봤는데, 더욱 길고 복잡한 프로그램을 디버깅 할 때 매우 유용할 것 같았다.