**시스템 소프트웨어 학기 프로젝트**

C프로그래밍 언어와 리눅스를 이용한 어셈블러 설계 및 개발

화 / 123 김광훈 교수님

#201410012

융합보안학과 오정규

설계이슈 1. 어셈블러의 아키텍처 결정

- Two-pass 어셈블러를 통해 첫 번째 패스에서 기호와 리터럴을 처리하고, 두 번째 패스에서 처리한 프로그램을 목적 프로그램으로 만들 것 이다. One-pass 어셈블러는 변수 등 각종 리터럴을 아래쪽에 사용할 경우 전방참조에 의한 오류가 발생할 수 있으므로 Two-pass 어셈블러를 통해 우선 처음에 코드를 쭉 훑음으로써 전방참조에 대한 정보를 참고하고, 그 정보를 두번째 패스를 돌리면서 활용함으로써 전방참조로 인해 발생할 수 있는 문제점을 해결 할 수 있다. 그러므로 Two-pass 어셈블러를 아키텍처로 결정 하였다.

설계이슈 2. Micro-Instruction Set Table 데이터 구조 결정

- Array 즉 배열을 통해 명령어들을 저장하고 사용하도록 할 것이다. 각종 명령어들을 배열에 저장해 두고, 그 배열의 명령어를 파악하여 case문 등을 사용해 명령어 별로 동작되는 행위들을 명령어와 동명의 함수를 생성해 처리하도록 만들 것이다.

설계이슈 3. Micro-Instruction Set Table의 검색 알고리즘 결정

- 코드의 명령어에 해당하는 부분들을 배열에 저장되어 있는 명령어들과 비교하여 해당하는 명령어를 찾도록 만들고, 그 명령어의 기능을 수행 할 수 있도록 한다.

설계이슈 4. 어셈블리언어상의 Addressing Mode 표기 방법 설정

- 주소를 표기하는 형식은 int형 배열을 선언하여 START명령어로 설정된 시작 지점을

int array[0]번에 놓고, 0번 이후의 배열의 값들은 기본 단위인 1워드 즉 3바이트씩 증가한다.

3형식과 4형식을 구분 하는 방법은 n,i,x,p,b,e의 플래그 비트를 이용한다.

설계이슈 5. 어셈블리언어상의 Directive 처리 방법을 결정한다.

- 어셈블리언어의 명령어들은 교재의 뒤쪽 부록에 쓰여있는 이름을 기본적으로 사용하거나, 나의 편의성을 위해 부록의 명령어들보다 더 명렁어의 행위를 잘 함축하고 있는 단어를 사용할 수 있도록 처리 할 것이다.

설계이슈 6. 목적 프로그램의 구조를 결정한다.

- 목적 프로그램은 교재의 어셈블러의 목적 프로그램과 같이 H, T, E 부분과 XE버전에 추가로 되어있는 E 부분으로 구성하도록 할 것이며, 목적 프로그램의 가독성을 조금이라도 향상시키기 위해 공백이나 다른 기호 등으로 OPCODE들을 구분하여 출력할 수 있도록 할 것이다.