

M68020 C CROSS COMPILER SYSTEM의 3B20S에의 이식

김완태* 정상현* 최영철** 유근호** 여제홍***

*승실대학교 전자계산학과 **육군사관학교 전자공학과 ***동양전자통신주식회사

PORTING OF M68020 C CROSS COMPILER SYSTEM
ONTO 3B20S COMPUTER

Wan-Tae Kim* Sang-Hyun Jeoung* Young-Cheal Choe** Keun-Ho Ryoo** JaeHeung Yuh***

*Dept. of Computer Science, Soong-Sil Univ. **Dept. of Elect., Korea Military Academy.
***Oriental Telecommunication Company, LTD

ABSTRACT

This paper has been aimed to develop M68020 Software Development System on a host computer 3B20S for the TDX-2 fully electronic Switching system by transporting M68020 C Cross Compiler.

M68020 C Cross Compiler source code which includes assembler, run-time library and optimizer has been analyzed for the installation on the host computer 3B20S system. Moreover, the linkage editor source file has been analyzed and installed on the 3B20S to produce the executable file correctly. Through these procedures, the M68020 object codes could be obtained on the 3B20S computer for the multi-using purposes. It has also been confirmed that the M68020 Software Development System on the 3B20S works correctly.

1. 서론

현대 사회는 정보를 수집하고 처리하며 분배하는 일이 급증하게 되었으며, 디지털 전송 및 교환 기술이 비약적으로 발전함에 따라 종래의 음성 신호 뿐만 아니라 Video, Facsimile 등 다양한 서비스를 제공할 수 있도록 하기 위하여 마이크로 컴퓨터의 발전에 크게 의존하고 있다. 이와 같이 정보의 가치가 급증하고 다양한 서비스를 제공하기 위하여 디지털 전송 및 교환 기술의 발전은 마이크로프로세서의 이용에 크게 의존하며, 마이크로 프로세서의 소프트웨어 개발 역시 급속하다.

M68020은 Motorola에서 개발한 32bits 마이크로 프로세서로서 현재 마이크로 컴퓨터의 CPU로 많이 사용되고 있으며 앞으로 극산 전자교환기 개발에 중요한 역할을 담당할 것으로 기대된다. 그러나 마이크로 컴퓨터는 소프트웨어 개발환경이 좋지 못하므로, 풍부한 소프트웨어와 큰 기억공간, 빠른 처리속도 및 다수 사용자의 시스템 이용 등의 이점을 지닌 중형 시스템인 3B20S에서 M68020의 object code를 생성케 하는 것은 소프트웨어 생산성을 높이는 첩경이 될 것이다. 그러므로 본 논문에서는 M68020 object code를 얻기 위하여 C Cross Compiler를 3B20S의 UNIX SYSTEM V에 이식하여, cc20 Compiler, optim20 Optimizer, as20 Assembler 등의 도구를 생성하였으며, 이에 필요한 Linker Editor System을 이식하여 M68020 소프트웨어 개발 환경을 구축하는 것이 본 논문의 목적이다.

2. 시스템 개요

2.1 C Cross Compiler 시스템 [4]

이 system은 M68020 cross C compilation system release 2, version 2.0 으로서 M68000 software generation system (SGS)으로 부터 유래했으며 AT&T Technologies Inc.와 Motorola Inc.가 공동개발한 것으로 t-

arget machine으로 Benchmark /20 을 위한 것이다.

이 M68020 cross C compilation system을 실행할 수 있는 경우는 system V/68을 실행하는 EXORmacs상에서, 15Mb 혹은 40Mb Winchester drive를 사용하는 VMR/10 system 상에서, 혹은 system V를 올린 VAX 11/780 상에서 실행할 수 있다.

2.2 Link Editor 시스템 [2]

Link Editor ld는 UNIX system 지원 tool로서 Motorola EXORmacs Development System과 VAX processor, 3B 계열의 모든 processor에 사용된다. Link Editor의 중요한 기능은 object file을 합치고 relocation을 수행하여 external reference를 해결하므로써 실행가능한 파일을 생성하는 것이다. Link Editor는 또한 symbolic debugging을 위한 정보를 갖고 있다.

또 사용자가 ld를 유연하고 정확하게 제어할 수 있도록 제어 언어(command language)를 이해하기도 하는데 이 제어 언어는 machine의 memory configuration을 나타낼 수 있고 object file을 특수한 형태로 조합하거나 file을 특정한 주소에 bound 하던가 memory의 특정한 portion 내에 있도록 하는 것과 link edit할 때에 global system을 정의하거나 재정의하는 일을 할 수 있도록 하고 있다. object file의 가장 작은 relocation 단위를 section 이라하고 memory의 연속 block이어야 한다. 또 object file은 assembler와 link editor에 의해 생성되고 입력으로 재배치 가능한 object file을 받아들이고 마찬가지로 재배치가 가능한 object file을 생성하던가 option으로 하지 않던가 할 수 있다. 그리고 compiler/assembler가 생성한 file은 언제나 세 개의 section .text .data .bss 로 구성된다.

3. 시스템 이식

3.1 개요

전체적인 porting 과정을 diagram으로 요약하면 그림 1과 그림 2로 나타낼 수 있다. 그림 1은 porting과정에서 발생할 수 있는 error에 대한 조치를 중심으로 나타낸 것이고 그림 2는 3B20S에 porting 후 사용할 수 있는 command인 cc20과 ld20을 이용하여 생성한 file.s, file.o, a.out을 비교하므로써 porting을 완료하고 그 결과를 검토하는 flow를 중심으로 나타낸 것이다.

3.2 C Cross compilation system의 이식

이 시스템의 구성은 5개의 디렉토리 bin, lib, src, usr, man으로 나눌 수 있다. 이 시스템을 이식하기 위하여 먼저 HOST 컴퓨터의 슈퍼 유저로 로그인하여야 한다. 그 후 working 디렉토리에서 제일 중심이 되는 makefile인 cross20.mk을 찾아서 install 부분을 실행시킨다. makefile에 나타나 있는 시스템 dependent한 변수를 찾아서 HOST인 3B20S에 적합하도록 수정하였고 유저가 필요로 하는 path도 HOST인 3B20S에 적합하도록 수정하였다.

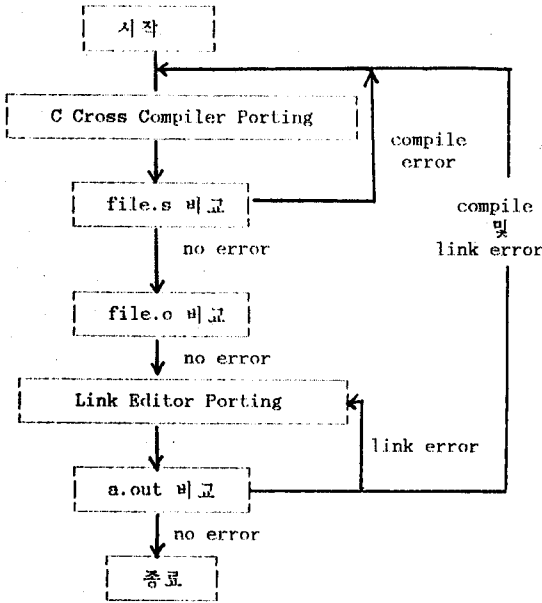


그림 1 porting 절차중심의 diagram

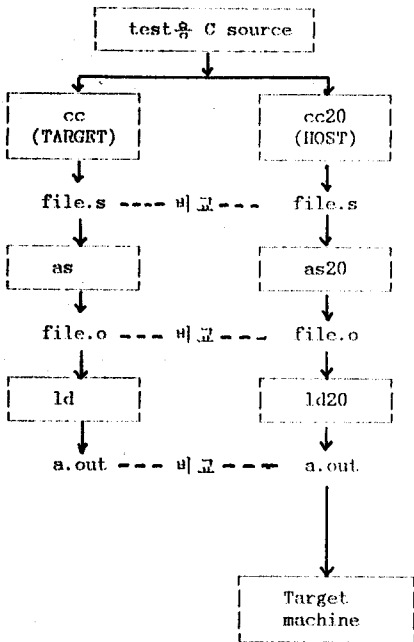


그림 2 생성되는 object-code 중심의 비교 diagram

이 system이 porting 된 후 생성되는 중요한 tool 로는

- cc20 compiler
- optim20 optimizer
- as20 assembler

로서 Motorola M68020 microprocessor 를 target 으로 한 object program을 생성한다. 특히 cc20 은 사용자와 interface 하는 M68020 Cross C Compiler 로서 이 system의 main driver routine이며 M68020 Assembly 명령어를 생성시킨다. 또 일단 cc20, as20

들이 구현된 다음에 M68020 Assembly Code를 cc20, as20을 이용하여 libb20.a 와 libb20.R1.a의 companion library 를 구성하고 ld20 (여기의 ld20은 Link Editor System 을 porting 후 생성된)을 이용해 a.out을 생성시킨다. 이 system을 3B20S의 UNIX SYSTEM V 에 installation 하여 M68020을 위한 software 개발 환경을 구축하기 위한 일련의 처리 과정을 그림 3 과 같이 도시 할 수 있다.

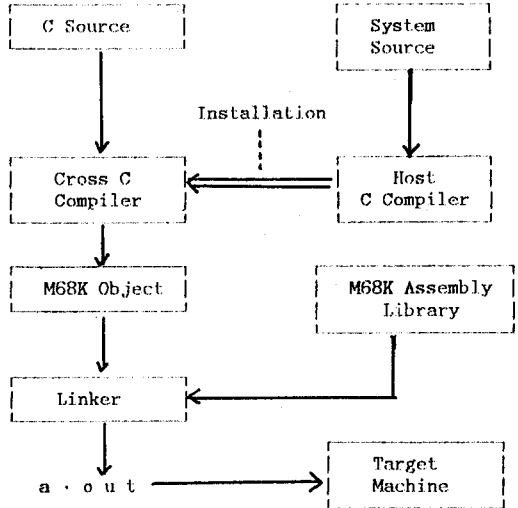


그림 3 C Cross Compiler을 PORTING하기 위한 과정

본 연구의 cross compilation 과정을 통하여 생성되는 cc20 cross compiler를 T diagram [1]을 통하여 나타내면 아래와 같다.

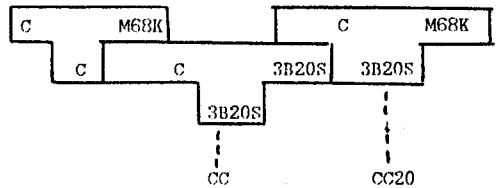


그림 4 cc20 Cross Compiler T diagram

3.3 LINK EDITOR SYSTEM 이식

Link editor 는 UNIX 시스템 지원 tool로서 Motorola EXORmacs Development system과 VAX processor, 3B 계열의 모든 프로세서에 사용된다. 이 시스템의 구성은 linker를 구성할 때 필요한 라이브러리 파일을 libld.a 를 구성하기 위한 libld 와 header file을 모아둔 inc, 그리고 linker source를 모아둔 ld로 디렉토리를 나눌 수 있다. linker 를 이식하는 이유는 3B20S 에 있는 linker command(ld)는 3B20S에서 실행가능한 오브젝트 코드를 생성하는 것이 목적이다. 그러나 cc20에서 생성된 코드는 3B20S 가 아닌 Target 컴퓨터에서 실행될 수 있는 코드를 생성하도록 하기 위해서는 Target 에 적합한 linker를 구성하여야 한다. 그러기 위해서는 C cross compilation의 이식과 마찬가지로 makefile에 대한 분석을 하였으며 link editor가 이식 될 HOST 를 지정 해주는 부분인 system.h 를 Target 에 맞게 고쳤다. 유저가 linker를 사용할 때 참조로 하는 path 를 지정하는 path.h 프로그램을 수정하였다[5].

COMMON OBJECT FILE FORMAT[3]은 cc20, as20, ld20 에 대해서 최종적으로 생성되는 실행 가능한 파일을 말한다. 이것은 링크과정에서 crt020.o, 라이브러리 파일, 오브젝트 파일들의 조합으로 이루어 지고 default20.ld 라고 하는 메모리 배치에 관한 정보를 가지고 각 section의 allocation address를 결정한다.

link editor의 source에서 main program은 lex.yy.c 에서 찾아 볼 수 있으며 그림 5 와 같은 function 을 call 하고 있다. ld/mc68/lex.yy.c 에서 main function이 call 하는 것은 link editor 의 pass one의 역할을 하고 있다. link editor의 source는 크게 pass one과 pass two로 나눌 수 있고 pass one은 link command line에서 option과 file name을 읽어 들이고 제어권을 pass two로 넘겨준다.

그림 5 에서 function inittpass 은 lex.yy.c에 있으며 pass one의 초기치 설정을 하는 곳이다. function pflags 는 link command line에서 읽어들인 option을 처리하는 것이고 마찬가지로 function filespec 는 filename 을 처리하는 곳이다. 그리고 pass2를 call하게 되면 제어는 ld01.c로 넘어간다.

```
main()
  inittpass1()  --> lex.yy.c
  pflags()     --> ld00.c
  filespec()   --> ld00.c
  pass2()      --> ld01.c
```

그림 5 main function 내에서 call 되는 function

본제점은 C cross compiler 의 main makefile 에서 ld (linkage editor)와 cpp(C preprocessor)는 HOST (여기서는 3B20S)를 이용하게 되어 있는데 실제 3B20S에서 cc20 으로 생성한 a.out file 과 M68020 에서 cc로 생성한 a.out file 과는 magic number가 서로 상이하여 일치하지 않았다. 따라서 3B20S 에서 생성한 a.out 과 M68020에서 생성한 a.out 을 서로 같은가를 비교하였다. 같은 a.out 이 생성되기 위해서는 link editor system의 어느 부분을 수정해야 하는가를 확인 한 후 수정하고 다시 a.out을 생성하고 확인하는 과정을 반복하였다.

3.4 요약 및 검토

지금까지 3B20S 에서 M68020 code 를 생성시키기 위하여 두개의 system, 즉 C Cross Compiler 와 Link Editor 를 porting 하기 위하여 makefile 및 source 의 적절한 path 의 지정과 parameter setting 을 통하여 cc20, as20, ld20 을 사용할 수 있도록 하였고 또 실행 가능한 file 인 a.out을 철저히 분석함으로써 a.out을 생성하고 실행하는데 발생한 error 를 고칠 수 있도록 하였다.

특히 그림 1 에서 크게 세가지의 error 에 대하여 다음과 같이 조치하였다.

compile error : file.s 와 file.o 를 비교해 본결과 상이한 점이 없었다.

link error: relocatable reference를 포함하고 있는 program을 실행시켰을 때 relocation type에 대한 예러가 발생하여 relocation type 에 관계된 source를 검토하여 switch (reentry.r_type) 부분의 flag 값이 024(octal) 일 경우에 symbol의 virtual address 에 대한 32-bit reference 를 수행해야 하는데 link editor source 에는 이 line 이 없다.

따라서 다음과 같이 case 문장을 추가하여 relocation 대한 예러를 제거하였다.

```
case R_PCRLONG: numbytes = 4;
```

compile 및 link error: cc20 명령시 아무런 option을 주지 않으면 직접 a.out을 생성하게 된다. 그런데 이렇게 생성된 a.out을 M68020에서 실행시켜 보니 실행이 되지 않아서로 비교하여 보니 optional header 부분이 누락되어 있었다. 그래서 cc20의 main source인 cc20.c program을 재분석 해보니 compile이 끝나고 link 명령 지정 (즉, link parameter setting) 과정에서 optional header 부분을 삽입 해주는 "X" option이 없었다.

따라서 cc20.c program의 link call routine의 시작 부분을 다음을 삽입시켰다.

```
av[na++] = "-X" ;
```

즉 ld20 명령을 따로 실행 할 때는 user가 X option을 주면 되지만 cc20으로 직접 a.out을 만들어 내고자 할 때는 program 내에서 X option을 지정해 주어야한다.

4. 결 론

본 논문은 이식성이 우수한 M68020 용 C Cross Compiler 를 3B20S 의 UNIX SYSTEM V 에 porting 하므로써 host 에서 M68020 machine code 생성을 위한 software 개발환경을 조성하였다.

이를 위하여 첫째, cross compiler의 source에 대한 분석, makefile 및 structure 에 대한 파악을 하였다. 특히 makefile에서 machine 에 관련된 부분을 가려내어 3B20S에 install 하였다.

둘째, 간단한 C program을 이용하여 3B20S에 생성된 object code와 M68020에서 생성된 object code 를 비교 하여 이상이 없음을 확인하고 전자교환기용 program, game program, benchmark program 등을 test 하여 마찬가지로 이상이 없음을 확인하였다.

셋째, 3B20S 에서 실행가능한 M68020 용 a.out을 생성하기 위하여 link editor source 을 구입하여 system parameter setting, main routine 및 directory 분석, editor를 porting하였다. 아울러 3B20S 및 M68020 a.out 의 file을 hexa dump하여 a.out의 구조와 내용을 비교분석하여 올바른 M68020 용 a.out이 생성 되도록 하였다.

내째, 이상의 과정을 통하여 M68020용 a.out을 host 인 3B20S 에서 얻을 수 있도록 software개발환경을 구성 하였다.

결론적으로 C Cross Compiler System과 Link Editor System 을 3B20S 에M68020용 software 개발 환경 구축을 완성 하였으며 본 논문의 연구결과 3B20S는 물론 다른 machine에도 install할 수 있는 기반을 조성하였다.

참고문헌

- [1] Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman compilers, principles, techniques, and tools Addison Wesley, 1986.
- [2] System V/68 Common Link Editor Reference Manual, Motorola Inc., Dec. 1985.
- [3] Technical Documentation IV : The Common Object File Format, Motorola Inc., Dec. 1985.
- [4] SGS M68020 Cross Compilation System Reference Manual, 2nd ed. Motorola Inc. 1985.