

CSE4100 시스템 프로그래밍

개별 프로젝트 #3

담당교수 : 서강대학교 컴퓨터공학과 김지환

1. 프로젝트 문제 및 목표

프로젝트 #1, #2 에서 구현한 셸(shell)에 linking과 loading 기능을 추가하는 프로그램입니다. 프로젝트 #2 에서 구현된 assemble 명령을 통해서 생성된 object 파일을 link시켜 메모리에 올리는 일을 수행합니다.

2. 요구사항

2.1 프로젝트 목표 설정

- 이미 제출한 프로젝트#2 에 아래의 기능들을 추가해야 합니다.
- 구현해야 할 사항들
 - ① 주소 지정 명령어 (progaddr)
 - ② Linking Loader (loader)
 - ③ 프로그램 실행 (run)
 - ④ debug 명령어 (bp)

2.2 합성

본 프로젝트 #2 에서 구현한 셸(shell)에 linking과 loading 기능을 추가하는 프로그램으로, 프로젝트 #2 에서 구현된 assemble 명령을 통해서 생성된 object 파일을 link시켜 메모리에 올리는 일을 수행한다. 주소 지정 명령어, Linking Loader, 프로그램 실행 명령어, debug 명령어 등을 구현해야 하는데 이를 위해 필요한 자료구조 및 알고리즘을 구상하여 전체 프로그램을 설계한다.

2.3 제작 / 2.4 시험

1) 주소 지정 명령어

sicsim> progaddr [address]

- loader 또는 run 명령어를 수행할 때 시작하는 주소를 지정합니다.
- sicsim이 시작되면 default로 progaddr는 0x00 주소로 지정됩니다.

시스템 프로그래밍 프로젝트 #3

ex) sicsim> progaddr 4000

- program address를 0x4000 주소로 지정합니다.
- loader 또는 run 명령어의 수행이 0x4000 부터 시작합니다.

2) Linking Loader

sicsim> loader [object filename1] [object filename2] [...]

- filename1, filename2, ... 에 해당하는 object 파일을 읽어서 linking 작업을 수행 후, 가상 메모리(1M)에 그 결과를 기록합니다.
(교재에 나오는 Pass1과 Pass2를 수행)
- Loader 실행이 성공적이면, load map을 화면에 출력합니다.
(교재 143P 참조)
- 에러가 존재할 경우, 에러 내용이 화면에 출력됩니다.

ex) sicsim> loader proga.obj progobj progobj

- proga.obj, progobj, progobj 를 가지고서 가상 메모리에 그 결과를 기록하고 load map을 화면에 출력합니다.

ex) sicsim> progaddr 4000

sicsim> loader proga.obj progobj progobj

control section	symbol name	address	length
<hr/>			
PROGA		4000	0063
	LISTA	4040	
	ENDA	4054	
PROGB		4063	007F
	LISTB	40C3	
	ENDB	40D3	
PROGC		40E2	0051
	LISTC	4112	
	ENDC	4124	
<hr/>			
total length			0133

sicsim> dump 4000, 4133

```
4000 .....
4010 .....
.....
4130 000083
```

3) 프로그램 실행

sicsim> run

시스템 프로그래밍 프로젝트 #3

- loader 명령어의 수행으로 메모리에 load된 프로그램을 실행합니다.
(프로그램의 구성에 I/O 명령어는 제외합니다.)
- progaddr 명령어로 지정한 주소부터 실행됩니다.
- 실행 결과로써 register 상태를 화면에 출력합니다.
출력되는 register는 A, X, L, PC, B, S, T 입니다.
- Breakpoint까지 실행되고 Breakpoint가 없으면 프로그램 끝까지 실행됩니다.
(명령어 bp 참조)

ex) sicsim> progaddr 4000

sicsim> loader proga.obj progb.obj progc.obj

control section	symbol name	address	length
<hr/>			
PROGA		4000	0063
	LISTA	4040	
	ENDA	4054	
PROGB		4063	007F
	LISTB	40C3	
	ENDB	40D3	
PROGC		40E2	0051
	LISTC	4112	
	ENDC	4124	
<hr/>			
total length			0133

simsim> run (주의: 아래 나타나는 register의 값들은 예시일 뿐임)

A : AAAABB X : 8C44D2

L : BBCCCC PC: 104134

B : FFFFFFFF S : A32B31

T : EEEEEEE

End program.

4) debug 명령어

sicsim> bp [address]

- sicsim에 breakpoint를 지정합니다.
Breakpoint는 bp 명령어를 통해서 무제한으로 지정할 수 있습니다.
- run을 수행하면 breakpoint까지 프로그램이 실행되고 프로그램이 정지합니다.
다음 번 run의 실행은 정지된 breakpoint부터 시작하고 그 다음 breakpoint까지 진행됩니다. Breakpoint가 없으면 프로그램 끝까지 실행됩니다.

sicsim> bp clear

- sicsim에 존재하는 breakpoint를 전부 삭제합니다.

시스템 프로그래밍 프로젝트 #3

sicsim> bp

- sicsim에 존재하는 breakpoint를 전부 화면에 출력합니다.

ex) sicsim> bp 4010

[ok] create breakpoint 4010

sicsim> bp 4020

[ok] create breakpoint 4020

sicsim> bp 4030

[ok] create breakpoint 4030

sicsim> bp

breakpoint

4010

4020

4030

sicsim> bp clear

[ok] clear all breakpoints

ex) sicsim> bp 4010

sicsim> bp 4020

sicsim> progaddr 4000

sicsim> loader proga.obj prog.b.obj prog.c.obj

simsim> run

A : AAAABB X : 8C44D2

L : BBCCCC PC: 104134

B : FFFFFFFF S : A32B31

T : EEEEEEE

Stop at checkpoint[4010]

simsim> run

A : AAAABB X : 8C44D2

L : BBCCCC PC: 104134

B : FFFFFFFF S : A32B31

T : EEEEEEE

Stop at checkpoint[4020]

simsim> run

A : AAAABB X : 8C44D2

L : BBCCCC PC: 104134

B : FFFFFFFF S : A32B31

T : EEEEEEE

End Program

시스템 프로그래밍 프로젝트 #3

```
simsim> run
A : AAAABB X : 8C44D2
L : BBCCCC PC: 104134
B : FFFFFFFF S : A32B31
T : EEEEEEE
Stop at checkpoint[4010]
```

2.5 평가

만점: 100 점, 보너스 추가 점수 40점

p145,146 에 있는 figure 3.12의 소스를 linking load 할 수 있어야 됩니다.
이 프로그램의 빈자리에 있는 코드들(즉, 빈자리의 text record들)은 무시해도 됩니다.

★ 배점표

	내용	배점
1단계	주소지정 명령어 와 Linking Loader 까지 (progaddr, loader)	100점
2단계	프로그램 실행과 debug 명령어 까지 (run, bp)	140점

3. 기타

3.1 환경 구성

Linux (gcc) : 반드시 gcc를 이용해서 C언어로 프로그램 하십시오.

특히 C언어가 아닌 C++ 등 다른 언어를 사용하거나, 도스 및 윈도우에서 작성한 경우 0점 처리합니다.

참고) 컴파일 시, make 파일에 gcc -Wall 옵션을 사용하여 warning 을 철저히 확인 하시기 바랍니다. (Warning 발생시 감점 처리함.)

3.2 팀 구성

개별 프로젝트입니다.

3.3 수행 기간: 4월 24일(일) 23:59까지

Late는 1일당 10%씩 감점. 최대 5일간만 인정함.

3.4 제출물 (5가지 파일중 하나라도 없는 경우에는 0점 처리함)

- 1) 프로그램 소스 및 헤더파일
- 2) Makefile
- 3) 프로그램 다큐멘테이션 리포트: 소스 및 프로그램의 구현방법을 설명한 Document. 반드시 예제 파일에 준해서 작성할 것 (제출하지 않거나 엉터리로 작성할 경우 최대 30% 감점)
- 4) 프로그램의 컴파일 방법 및 실행방법에 대한 간단한 내용을 적은 README파일
- 5) 테스트 파일 (ex) prog.a.obj, prog.b.obj, prog.c.obj, copy.obj)

3.5 제출 방법 (형식을 지키지 않을 경우 감점 처리함)

sp<학번>_proj3 이름의 디렉터리를 만들고, 여기에 위에서 설명한 모든 파일들을 넣은 후, 디렉터리를 tar로 압축하여 한 파일로 만들어 메일로 보내시기 바랍니다. (압축파일 내에 반드시 디렉터리가 포함되어 있어야 하며, 바이너리파일 및 코어파일을 제외할 것. 기타 불필요한 파일을 포함시키지 말 것.)

ex) sp20131234_proj3/

README → 컴파일 방법 및 실행방법에 대한 간단한 내용을 적은 파일
Document.doc → (또는 Document.hwp)
opcode.txt → 프로젝트#2에서 제공된 opcode 파일.
20131234.c → 소스 파일이 여러 개인 경우 main 함수가 있는 파일의 이름을 학번.c 로 합니다.
20131234.h → 최소 한 개 이상의 헤더 파일. 하나인 경우 학번.h.
Makefile → 실행파일은 20131234.out처럼 학번.out 이름으로 고정할 것.
proga.obj, prog.b.obj, prog.c.obj → linking loader에 대한 example.
copy.obj → run 명령어에 대한 example.

tar 명령어는 아래와 같이 사용합니다.

tar 파일로 묶을 때 지난 project와 동일하게 -z 옵션을 사용하지 않습니다.

ex) tar cvf sp<학번>_proj3.tar 만든 디렉토리명

tar 파일의 이름은 다음과 같이 되어야 합니다.

sp<학번>_proj3.tar

ex) sp20131234_proj3.tar

제출 주소 : sp2016proj@gmail.com

메일제목 형식 : [SP 숙제#3]학번 이름

(예: [SP 숙제#3]20131234 홍길동)

시스템 프로그래밍 프로젝트 #3

주의사항

메일로 첨부할 파일이 잘 작성되었는지 확인하고 보내시기 바랍니다.

+ 제출형식(메일제목, tar file 이름 형식, 내용물)이 잘못되었을 시, 감점 10%

+ 중복으로 메일 보낼 시, 1회에 추가로 5%씩 감점

+ 제출 시간이 늦춰질 시, 감점

24시간(1일) 이내 10%감점

2일 이내 20%감점

3일 이내 30% 감점

4일 이내 40% 감점

5일 이내 50% 감점

3.6 Source code 관련

Segmentation fault

실행 불가 시 : 0점

명령 수행 시 : 그 부분점수 0점

Warning

1점 감점

Average case

기본 예제 파일 수행

Boundary case

기본 예제 파일 수행 이외에 더 많은 것을 수행

1건당 5점 가산

주석

주석이 없거나, 알아볼 수 없는 경우 감점 시키겠습니다.

타인이 알아볼 수 있는 형태로 주석을 달아주십시오.

3.7 프로젝트에 대한 질문사항은 사이버캠퍼스(eclass)의 질의 게시판을 이용해 주세요.

3.8 남의 프로그램을 보거나 일 부분이라도 copy해서 제출한 사람은 이 과목의 기말성적이 F가 나갈 것임.