

샘플 컴파일러 CBU의 컴파일 및 수행 방법

2023. 4.

이재성/충북대 소프트웨어학부

1. CBU 컴파일러 만들기

환경 설정:

win_flex와 win_bison을 미리 설치하고 win_flex와 win_bison이 수행 가능한 폴더에서 아래 단계를 수행한다.

1단계: CBU 어휘분석기 소스 생성

```
C>win_flex --wincompat -o cbu2lex.c cbu2.l
```

win_flex가 cbu2.l 파일을 입력으로 받아 어휘분석기 파일 cbu2lex.c를 생성함

2단계: CBU 파서 소스 생성

```
C>win_bison -o cbu2.c -d cbu2.y
```

win_bison이 cbu2.y 파일을 입력으로 받아 구문분석기 파일 cbu2.c를 생성하고 또한 헤더파일 cbu2.h를 생성한다(-d 옵션이 헤더파일 생성 옵션이며, cbu2lex.c에서 이 헤더 파일을 사용)

3단계: CBU 컴파일러 소스(cbu2.c, cbu2lex.c)의 컴파일

Visual C++ 에서 cbu2.c, cbu2.h, cbu2lex.c를 프로젝트로 등록하여 컴파일한다. 컴파일 결과 나온 수행 파일이 바로 cbu 컴파일러(cbu2.exe)가 된다.

2. CBU 컴파일러 수행 테스트

1단계: cbu 언어 샘플 프로그램 컴파일

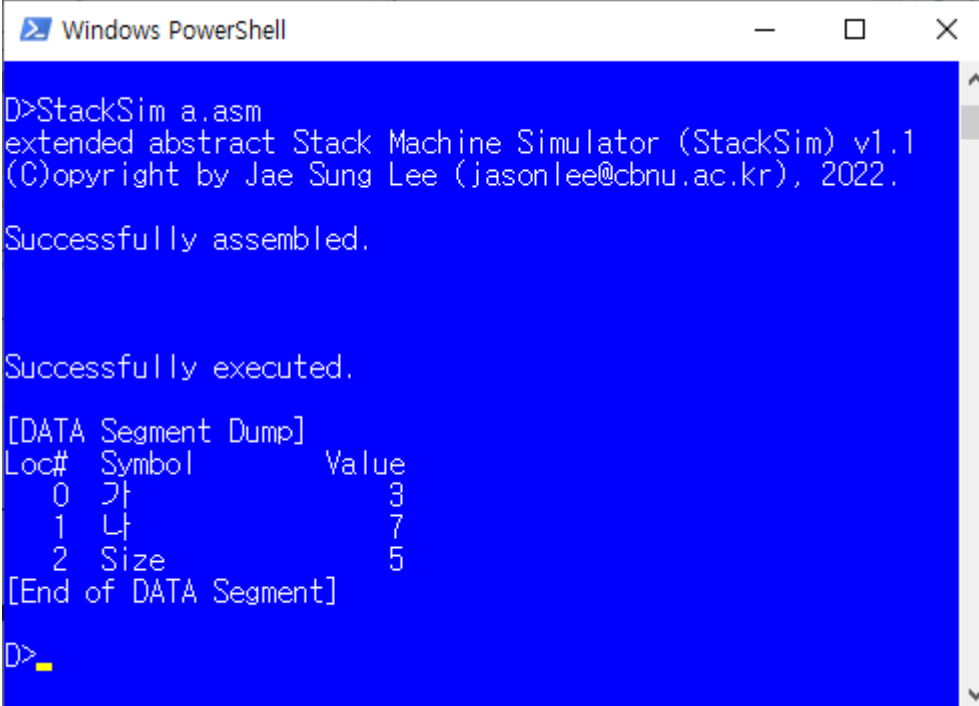
제공된 sample program (CBU언어로 작성된 프로그램)인 sample.cbu를 컴파일하여 수행해 본다. 실행 결과, 어셈블리어인 a.asm파일이 출력으로 생성된다.

```
C> cbu2 sample.cbu
```

2단계: 어셈블리 시뮬레이터 수행

스택기계 시뮬레이터 (StackSim.exe)를 이용하여 a.asm을 수행하여 결과를 확인한다.

```
C> StackSim a.asm
```



```
D>StackSim a.asm
extended abstract Stack Machine Simulator (StackSim) v1.1
(C)opyright by Jae Sung Lee (jasonlee@cbnu.ac.kr), 2022.

Successfully assembled.

Successfully executed.

[DATA Segment Dump]
Loc#  Symbol      Value
  0   가          3
  1   나          7
  2   Size         5
[End of DATA Segment]

D>
```

주의사항: 여기에서 제공되는 샘플 CBU 컴파일러는 대략적인 컴파일러 작성 흐름을 보여주기 위한 것이므로, 매우 간단히 작성되었다. 따라서 기능을 확장하기 위해서는 많은 수정이 필요하다.