

## 실습 3. Y86-64 시뮬레이터 – YAS와 YIS

---

순천향대학교 컴퓨터공학과

이 상 정

---

### 3.1 Y86-64 시뮬레이터 설치

## Y86-64 프로세서 시뮬레이터

### □ x86-64의 간략화 버전인 Y86-64 시뮬레이터

- YAS
  - Y86-64 어셈블러
- YIS
  - Y86-64 명령어 집합 시뮬레이터
- HCL2
  - HCL을 C로 변환
- SEQ, SEQ+
  - 순차적 Y86-64 프로세서 시뮬레이터
- PIPE
  - 파이프라인 Y86-64 프로세서 시뮬레이터

## Tcl/Tk 설치

### □ GUI 버전 시뮬레이터를 위해 Tcl/Tk 설치

- Tcl (Tool Command Language)은 스크립트 언어
- Tk (widget toolkit)는 GUI 인터페이스 툴킷

### □ 우분투 설치

- 시뮬레이션 툴 환경에 맞춰 버전 8.5 설치

```
linux> sudo apt-get install tcl8.5-dev
```

```
linux> sudo apt-get install tk8.5-dev
```

```
Selecting previously unselected package tcl8.5-dev:amd64.
Preparing to unpack .../tcl8.5-dev_8.5.19-4_amd64.deb ...
Unpacking tcl8.5-dev:amd64 (8.5.19-4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
libtcl8.5:amd64 (8.5.19-4) 설정하는 중입니다 ...
tcl8.5 (8.5.19-4) 설정하는 중입니다 ...
tcl8.5-dev:amd64 (8.5.19-4) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
linux>
```

```
Selecting previously unselected package tk8.5-dev:amd64.
Preparing to unpack .../tk8.5-dev_8.5.19-3_amd64.deb ...
Unpacking tk8.5-dev:amd64 (8.5.19-3) ...
libtk8.5:amd64 (8.5.19-3) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
tk8.5 (8.5.19-3) 설정하는 중입니다 ...
tk8.5-dev:amd64 (8.5.19-3) 설정하는 중입니다 ...
linux>
```

## Flex와 Bison 설치

### □ Flex (fast lexical analyzer generator)는 어휘 분석기

- `linux> sudo apt-get install flex`

```
Processing triggers for libc-bin (2.27-3ubuntu1) ...
libfl2:amd64 (2.6.4-6) 설정하는 중입니다 ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
flex (2.6.4-6) 설정하는 중입니다 ...
libfl-dev:amd64 (2.6.4-6) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
linux>
```

### □ Bison 은 구문 분석기

- `linux> sudo apt-get install bison`

```
Preparing to unpack .../bison_2%3a3.0.4.dfsg-1build1_amd64.deb ...
Unpacking bison (2:3.0.4.dfsg-1build1) ...
libbison-dev:amd64 (2:3.0.4.dfsg-1build1) 설정하는 중입니다 ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
bison (2:3.0.4.dfsg-1build1) 설정하는 중입니다 ...
update-alternatives: using /usr/bin/bison.yacc to provide /usr/bin/yacc (yacc) in auto mode
```

## 시뮬레이터 다운로드

### □ 가상머신에서 시뮬레이터 다운로드

- `linux> wget http://csapp.cs.cmu.edu/3e/sim.tar`

```
linux> wget http://csapp.cs.cmu.edu/3e/sim.tar
--2019-06-09 22:55:16-- http://csapp.cs.cmu.edu/3e/sim.tar
Resolving csapp.cs.cmu.edu (csapp.cs.cmu.edu)... 128.2.100.230
Connecting to csapp.cs.cmu.edu (csapp.cs.cmu.edu)|128.2.100.230|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 624640 (610K) [application/x-tar]
Saving to: 'sim.tar'

sim.tar          100%[=====] 610.00K   236KB/s   in 2.6s
2019-06-09 22:55:21 (236 KB/s) - 'sim.tar' saved [624640/624640]
```

```
linux> ls
ch3          share      공개      문서      비디오   음악
examples.desktop  sim.tar   다운로드  바탕화면  사진     템플릿
linux>
```

## 시뮬레이터 압축 해제

### □ 압축 해제

- `linux> tar xf sim.tar`

```
linux> tar xf sim.tar
linux> ls
ch3          share  sim.tar  다운로드  바탕화면  사진  템플릿
examples.desktop  sin    공개    문서      비디오    음악
linux> ls sim
Makefile  README  misc  pipe  ptest  seq  y86-code
linux>
```

## Makefile 수정

### □ Makefile 수정

- `linux> cd sim`
- `linux> nano Makefile`
  - 라인 3, 주석 해제, `GUIMODE=-DHAS_GUI`
  - 라인 10, 다음과 같이 수정  
`TKLIBS=-L/usr/lib -ltk -ltcl`  
 $\Rightarrow$  `TKLIBS=-L/usr/lib -ltk8.5 -ltcl8.5`

```
# Comment this out if you don't have Tk
GUIMODE=-DHAS_GUI

# Modify the following line so that gcc
# libtk.so libraries on your system. You
# to tell gcc which directory to look
# don't have Tcl/Tk.

TKLIBS=-L/usr/lib -ltk8.5 -ltcl8.5
#TKLIBS=-L/usr/lib -ltk -ltcl

# Modify the following line so that gcc
# header files on your system. Comment
# Tcl/Tk.

TKINC=-isystem /usr/include/tcl8.5
```

## matherr 에러 수정

- ❑ 시뮬레이터의 옛 버전 라이브러리로 인해 **matherr** 참조 에러 발생
- ❑ **matherr** 참조 주석 처리
  - seq/ssim.c, line 844
  - pipe/psim.c, line 809

```
#ifdef HAS_GUI

/*****
 * Begin Part 3 globals
 *****/

/* Hack for SunOS */
//extern int matherr();
//int *tclDummyMathPtr = (int *) matherr;

static char tcl_msg[256];
/* Keep track of the TCL Interpreter */
static Tcl_Interp *sim_interp = NULL;
```

## 시뮬레이터 설치

- ❑ 설치
  - **linux> make clean**
  - **linux> make**

```
make[1]: Entering directory '/home/arch/sim/y86-code'
../misc/yas abs-asum-cmov.yas
../misc/yas abs-asum-jmp.yas
../misc/yas asum.yas
../misc/yas asumr.yas
../misc/yas asumi.yas
../misc/yas cjr.yas
../misc/yas j-cc.yas
../misc/yas poptest.yas
../misc/yas pushquestion.yas
../misc/yas pushtest.yas
../misc/yas prog1.yas
../misc/yas prog2.yas
../misc/yas prog3.yas
../misc/yas prog4.yas
../misc/yas prog5.yas
../misc/yas prog6.yas
../misc/yas prog7.yas
../misc/yas prog8.yas
../misc/yas prog9.yas
../misc/yas prog10.yas
../misc/yas ret-hazard.yas
make[1]: Leaving directory '/home/arch/sim/y86-code'
linux>
```

```
make[1]: Leaving directory '/home/arch/sim/seq'
(cd y86-code; make all)
make[1]: Entering directory '/home/arch/sim/y86-code'
../misc/yas abs-asum-cmov.yas
../misc/yas abs-asum-jmp.yas
../misc/yas asum.yas
../misc/yas asumr.yas
../misc/yas asumi.yas
../misc/yas cjr.yas
../misc/yas j-cc.yas
../misc/yas poptest.yas
../misc/yas pushquestion.yas
../misc/yas pushtest.yas
../misc/yas prog1.yas
../misc/yas prog2.yas
../misc/yas prog3.yas
../misc/yas prog4.yas
../misc/yas prog5.yas
../misc/yas prog6.yas
../misc/yas prog7.yas
../misc/yas prog8.yas
../misc/yas prog9.yas
../misc/yas prog10.yas
../misc/yas ret-hazard.yas
make[1]: Leaving directory '/home/arch/sim/y86-code'
linux>
```

## 시뮬레이터 서브디렉토리

---

### □ sim 디렉토리 아래의 서브디렉토리

- misc
  - 유틸리티
  - YAS, YIS, HCL2C
- seq
  - SEQ, SEQ+ 시뮬레이터
- pipe
  - PIPE 시뮬레이터
- y86-code
  - 어셈블리 예제 프로그램
- ptest
  - 명령어들의 회귀분석 테스트 스크립트

---

## 3.2 YAS (Y86-64 assembler) / YIS (Y86-64 Instruction Simulator)

## 덧셈 예제 C 프로그램

```
#include <stdio.h>

long add(long *x, long y)
{ return *x+y; }

int main()
{
    long a, b, c;

    a = 2;
    b = 3;
    c = add(&a, b);

    return 0;
}
```

## 덧셈 Y86-64 어셈블리 프로그램 - add.y8

```
# Array of 4 elements
    .align 8
a:   .quad 2
b:   .quad 3
c:   .quad 0

main:
    irmovq a, %rdi    # &a
    mrmovq b, %rsi    # b
    call add          # add(&a, b)
    rmmovq %rax, c    # c = ax
    ret
```

```
# long add(long *x, long y)
# x in %rdi, y in %rsi
add:
    mrmovq (%rdi), %rax # *x
    addq %rsi, %rax     # a+b
    ret                # Return
```

```
# Stack starts here and grows to
# lower addresses
    .pos 0x200
stack:
```



## YAS 어셈블

## □ YAS 어셈블

- `linux> ~/sim/misc/yas add.yo`
- `add.yo` 목적파일 생성

```

0x018:                                | # Array of 4 elements
                                | .align 8
0x018: 020000000000000000          | a:      .quad 2
0x020: 030000000000000000          | b:      .quad 3
0x028: 000000000000000000          | c:      .quad 0

0x030:                                | main:
0x030: 30f7180000000000000000      | irmovq a, %rdi      # &a
0x03a: 506f200000000000000000      | mrmovq b, %rsi      # b
0x044: 8058000000000000000000      | call add            # add(&a, b)
0x04d: 400f280000000000000000      | rmmovq %rax, c      # c = ax
0x057: 90                          | ret

                                | # long add(long *x, long y)
                                | # x in %rdi, y in %rsi
0x058:                                | add:
0x058: 5007000000000000000000      | mrmovq (%rdi), %rax # *x
0x062: 6060                        | addq %rsi, %rax     # a+b
0x064: 90                          | ret                # Return

                                | # Stack starts here and grows to lower addresses
0x200:                                | .pos 0x200
0x200:                                | stack:

```

15

## YIS 실행

## □ YIS 실행

- `linux> ~/sim/misc/yis add.yo`
- 실행 후 변경된 레지스터와 메모리 내용 출력

```

linux> ~/sim/misc/yis add.yo
Stopped in 11 steps at PC = 0x13.  Status 'HLT', CC Z=0 S=0 O=0
Changes to registers:
%rax:  0x0000000000000000      0x0000000000000005
%rsp:  0x0000000000000000      0x0000000000000200
%rsi:  0x0000000000000000      0x0000000000000003
%rdi:  0x0000000000000000      0x0000000000000018

Changes to memory:
0x0028: 0x0000000000000000      0x0000000000000005
0x01f0: 0x0000000000000000      0x000000000000004d
0x01f8: 0x0000000000000000      0x0000000000000013

```



### □ 시뮬레이터 가이드

- <http://csapp.cs.cmu.edu/3e/simguide.pdf>