

Computer Architecture – LAB 0

김원표

kwp@hallym.ac.kr

Introduction

■ Instructor

- 이정근(Jeong-Gun Lee)
- Office : 공학관 1306
- E-mail : jeonggun.lee@hallym.ac.kr
- Tel : 033-248-2312

■ TA

- 김원표(Won-Pyo Kim)
- Office : 공학관 A1409 (Operating System Lab)
- E-mail : kwp@hallym.ac.kr
- Tel : 033-248-2395

Introduction

■ Lab Session

- 본 수업은 어셈블리 프로그래밍을 배우고 실습시간 내에 각자 해당하는 것을 완료할 수 있도록 한다.
- 실습 중 완료하지 못한 부분은 과제로 진행한다.

■ Evaluation

- Lecture 70% (Mid-term : 30%, Final : 40%)
- Lab 30% (Assignments & Small project)

■ Attendance Policy

- 'F' if more than 4 unnotified absences.

■ Practicum Policy

- 과제 진행의 결과물은 스마트 캠퍼스 제출을 원칙으로 함
- 기간 내에 제출하지 못한 경우 조교의 이메일로 제출 (감점)
- 제출 과제의 컨닝 발견시 해당 학생의 이후 모든 과제는 자필 제출

■ SPIM

- SPIM은 MIPS 프로세서의 시뮬레이터
- 어셈블리 코드를 실행하기 위해 설계된 소프트웨어이다.
- 편집공간이 따로 존재하지 않고 작성된 코드파일을 동작시킬 수 있다.

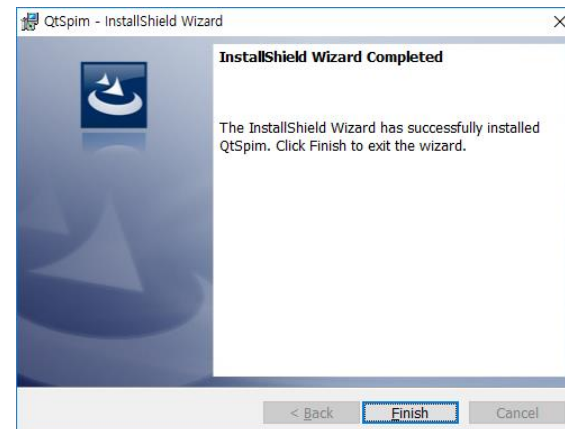
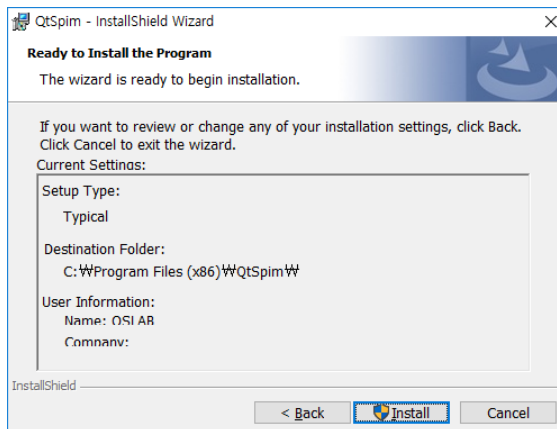
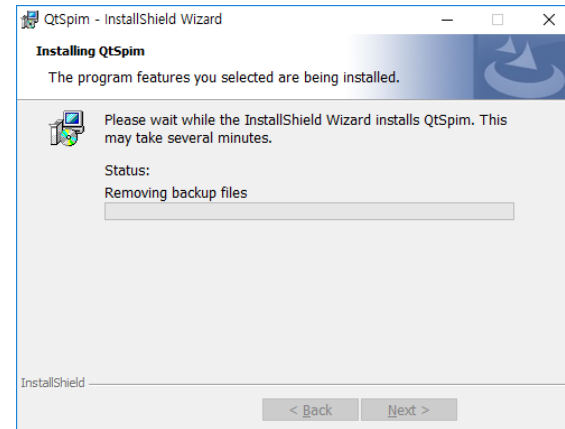
■ 실습에서는 QT-SPIM을 통해 실습을 진행한다(PC-SPIM을 사용해도 무방).

■ 설치

- http://sourceforge.net/projects/spimsimulator/files/QtSpim_9.1.12_Windows.exe/download
- or Smart Campus

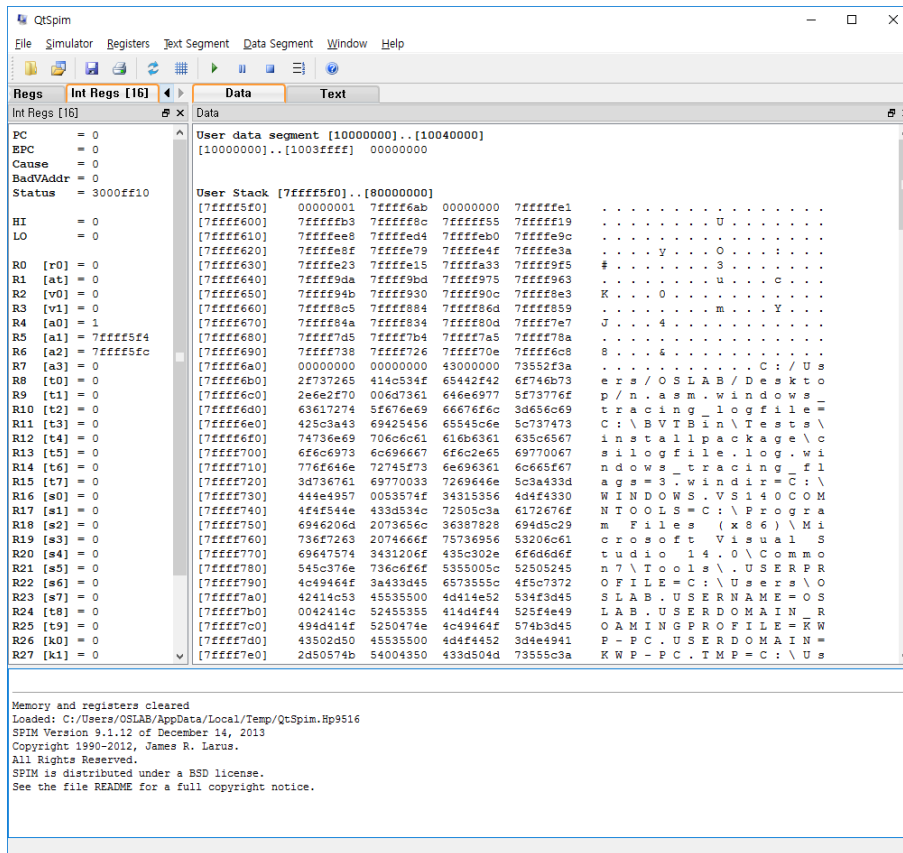
Setup

■ 설치과정



Setup

■ 초기 실행 모습



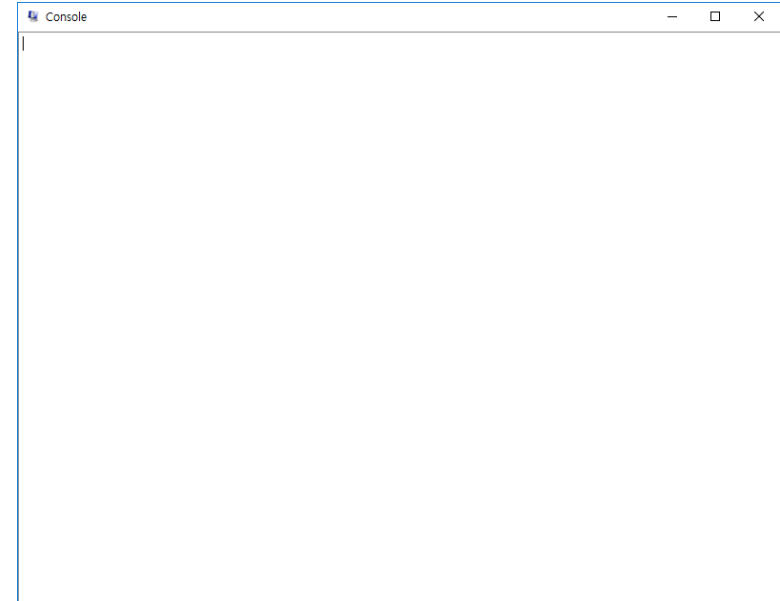
```
QtSpim
File Simulator Registers Text Segment Data Segment Window Help

Regs Int Regs [16] Data Text
Int Regs [16] Data
PC = 0
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000fff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 0
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 7ffff5f4
R6 [a2] = 7ffff5fc
R7 [a3] = 0
R8 [t0] = 0
R9 [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
R22 [s6] = 0
R23 [s7] = 0
R24 [t8] = 0
R25 [t9] = 0
R26 [k0] = 0
R27 [k1] = 0

User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000

User Stack [7ffff5f0]..[80000000]
[7ffff5f0] 00000001 7ffff6ab 00000000 7fffffe1 . . . . .
[7ffff600] 7fffffb3 7fffffb3 7fffffb3 7fffffb3 . . . . .
[7ffff610] 7fffffe8 7ffffed4 7ffffeb0 7ffffe9c . . . . .
[7ffff620] 7ffffe8f 7ffffe79 7ffffe4f 7ffffe3a . . . . .
[7ffff630] 7ffffe23 7ffffe15 7ffffa33 7ffff9f5 # . . . . .
[7ffff640] 7ffff9da 7ffff9bd 7ffff975 7ffff963 . . . . .
[7ffff650] 7ffff94b 7ffff930 7ffff90c 7ffff8e3 . . . . .
[7ffff660] 7ffff9c5 7ffff884 7ffff86d 7ffff859 . . . . .
[7ffff670] 7ffff94a 7ffff834 7ffff80d 7ffff7e7 . . . . .
[7ffff680] 7ffff7d5 7ffff7b4 7ffff7a5 7ffff78a . . . . .
[7ffff690] 7ffff738 7ffff726 7ffff70e 7ffff6e8 . . . . .
[7ffff6a0] 00000000 00000000 43000000 73552f3a . . . . .
[7ffff6b0] 2f737265 414c534f 65442f42 6f746b73 e r s / O S L A B / D e s k t o
[7ffff6c0] 2e6e2f70 006d7361 646e6977 5f73776f p / n . a s m . w i n d o w s =
[7ffff6d0] 63617274 5f676e69 66676f6c 3d656c69 t r a c i n g _ l o g f i l e =
[7ffff6e0] 425c3a43 69425456 65545c6e 5c737473 C : \ B V T B i n \ T e s t s \
[7ffff6f0] 74706e69 706c6c61 616b6361 635c6567 i n s t a l l p a c k a g e \ c
[7ffff700] 6f6e6973 6c696667 6f6e2e65 69770067 s i l o g f i l e . l o g . w i
[7ffff710] 77f6646e 72745f73 6e696361 6c656f67 n d o w s _ t r a c i n g _ f l
[7ffff720] 3d736761 69770033 7269646e 5c3a433d a g s = 3 . w i n d i r = C : \
[7ffff730] 444e4957 0053574f 34315356 4d4f4330 W I N D O W S . V S 1 4 0 C O M
[7ffff740] 4f4f544e 433d534c 72505c3a 6172676f N T O O L S = C : \ P r o g r a
[7ffff750] 6946206d 2073656c 36387828 694d5c29 m F i l e s ( x 8 6 ) \ M i
[7ffff760] 736f7263 2074666f 75736956 53206c61 c r o s o f t V i s u a l S
[7ffff770] 69647574 3431206f 435c302e 6f6d6d6f t u d i o 1 4 . 0 \ C o m m o
[7ffff780] 545c376e 736c6f6f 5355005c 52505245 n 7 \ T o o l s \ . U S E R P R
[7ffff790] 4c49464f 3a433d45 6573555c 4f5c7372 O F I L E = C : \ U s e r s \ O
[7ffff7a0] 42414e53 45535500 4d414e52 534f4945 S L A B . U S E R N A M E = O S
[7ffff7b0] 0042414c 52455355 414d4f44 525f4e49 L A B . U S E R D O M A I N = R
[7ffff7c0] 494d414f 5250474e 4c49464f 574b3d45 O A M I N G P R O F I L E = K W
[7ffff7d0] 43502d50 45535500 4d4f4452 3d4e4941 P - P C . U S E R D O M A I N =
[7ffff7e0] 2d50574b 54004350 433d504d 73555c3a K W P - P C . T M P = C : \ U s
```

Memory and registers cleared
Loaded: C:/Users/OSLAB/AppData/Local/Temp/QtSpim.Hp9516
SPIM Version 9.1.12 of December 14, 2013
Copyright 1990-2012, James R. Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.



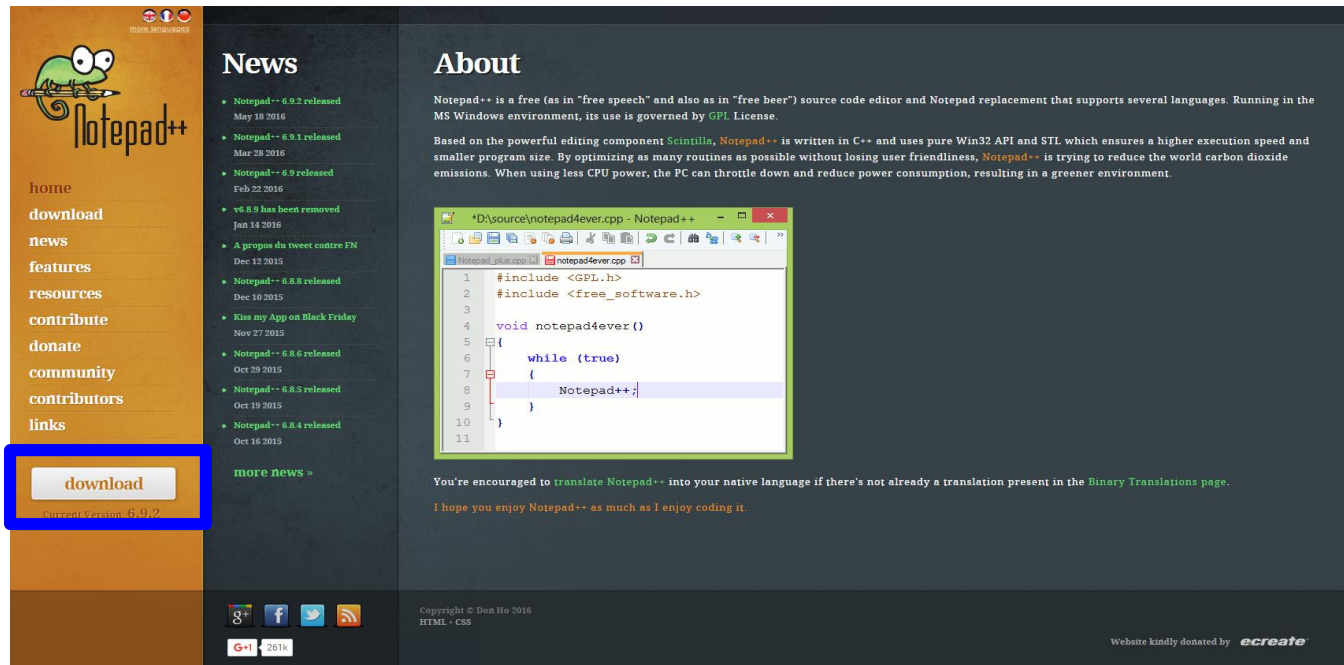
Setup

■ Notepad & Notepad++

- SPIM은 시뮬레이터로 별도의 작업공간(코딩)이 구비되어 있지 않다.
- 따라서 Notepad나 Notepad++로 코딩을 하여 파일을 저장한 후 시뮬레이팅을 하면된다.
- 둘 중 아무거나 본인이 편한 것을 사용하면 된다.

■ Notepad++

- <https://notepad-plus-plus.org/>



The screenshot shows the Notepad++ website. On the left, there is a navigation menu with links: home, download, news, features, resources, contribute, donate, community, contributors, and links. The 'download' link is highlighted with a blue box. Below the navigation menu, there is a 'download' button and the text 'Current Version: 6.9.2'. On the right, there is a 'News' section with a list of recent releases and an 'About' section with a description of Notepad++ and a code editor window showing a C++ program.

News

- Notepad++ 6.9.2 released May 18 2016
- Notepad++ 6.9.1 released Mar 28 2016
- Notepad++ 6.9 released Feb 22 2016
- v6.8.9 has been removed Jan 14 2016
- A propos du tweet entire FN Dec 12 2015
- Notepad++ 6.8.8 released Dec 10 2015
- Kiss my App on Black Friday Nov 27 2015
- Notepad++ 6.8.6 released Oct 29 2015
- Notepad++ 6.8.5 released Oct 19 2015
- Notepad++ 6.8.4 released Oct 16 2015

About

Notepad++ is a free (as in "free speech" and also as in "free beer") source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by [GPL License](#).

Based on the powerful editing component [Scintilla](#), Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.

```
#include <GPL.h>
#include <free_software.h>

void notepad4ever()
{
    while (true)
    {
        Notepad++;
    }
}
```

You're encouraged to [translate Notepad++](#) into your native language if there's not already a translation present in the [Binary Translations](#) page.

I hope you enjoy Notepad++ as much as I enjoy coding it.

Copyright © Don Ho 2016
HTML - CSS

Website kindly donated by [ecreate](#)

Setup

■ Notepad++

- 원하는 형식(설치형, 포터블형)을 선택

Home > Download > v6.9.2 - Current Version

Download Notepad++ 6.9.2

Tweet Like 921 +1 1.1k

Intel® Modern Code software.intel.com/modern-code Multi-Level Parallel Algorithms that Effectively Scale Hardware.

온라인 성서히브리어 학습 - 성경을 깊게 이해하기
lo.teacherbiblical.com/성경+히브리어
예루살렘 히브리 대학교 인증

내 취향저격 게임은?
orakail.withgoogle.com
Google Play 오락실에서 세상에 없던 플레이도 찾아보세요

AJ렌터카 공식 사이트 - 제주 AJ렌터카
ajrentacar.co.kr
AJ렌터카, 제주도 여를 성수기 EVENT! SUV, 고급차, 승합차 최대 80% 특별할인.

Release Date: 2016-05-18

DOWNLOAD

Notepad++ Installer: Take this one if you have no idea which one you should take.

- Notepad++ zip package: Don't want to use installer? Check this one (zip format).
- Notepad++ 7z package: Don't want to use installer? 7z format.
- Notepad++ minimalist package: No theme, no plugin, no updater, quick download and play directly. 7z format.
- SHA-1 digests for binary packages: Check it if you're paranoid.
- Notepad++ source code: The source code of current version.

You can find sha1 digests for binaries package on Github as well.

Notepad++ v6.9.2 new features and bug-fixes:

1. Add most wanted feature: Log Monitoring (tail -f).
2. Add new feature: Find in Find...

download

eTeacherBIBLICAL®
Online Language Academy

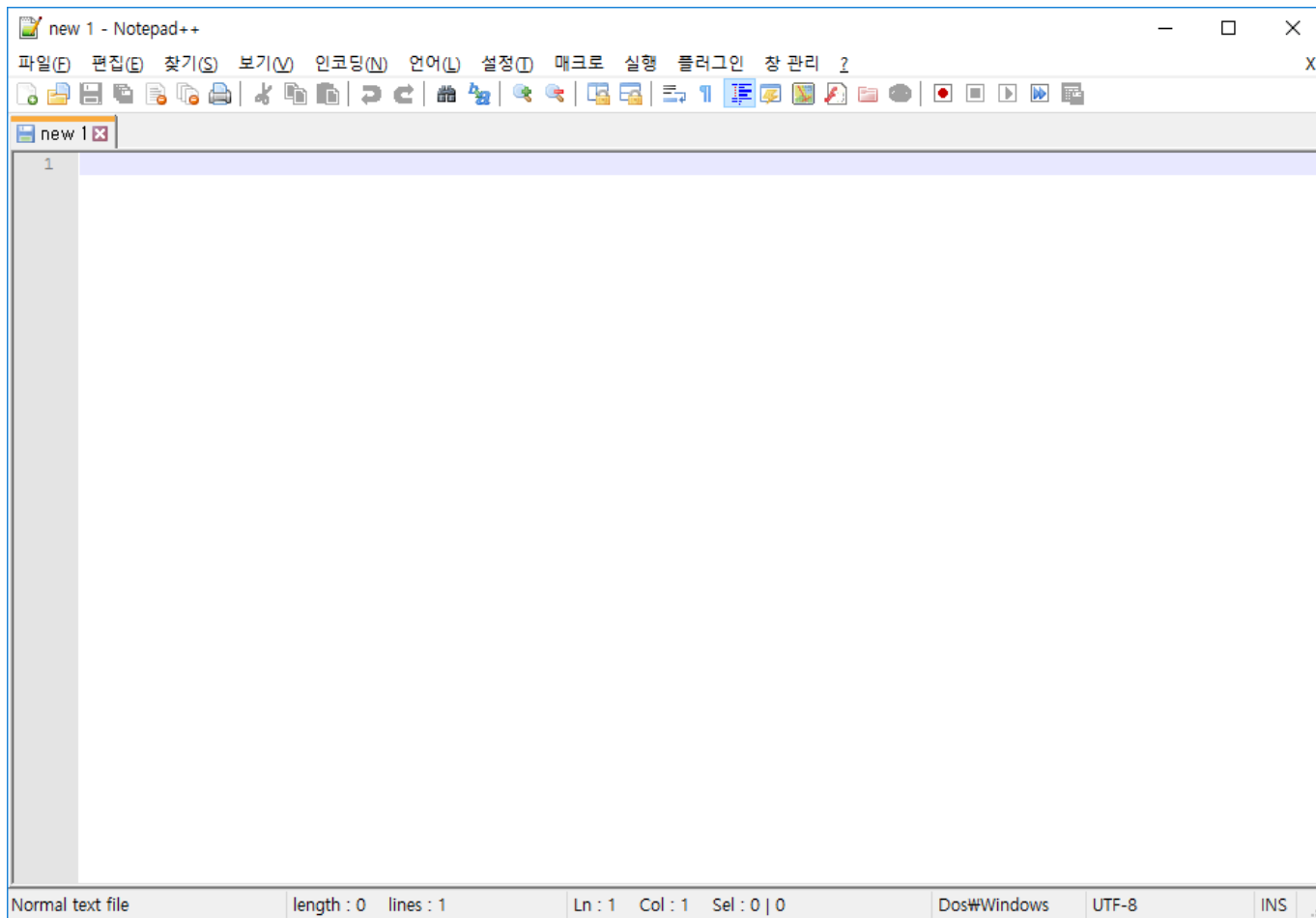
성서히브리어를
온라인으로 배우세요
성경에 대한 깊은 이해를 위해
지금 신청하세요!

놀라울 정도로
새로워졌습니다!
Creative Cloud의 새로운 기능을 확인하세요.
플랜 비용은 월 ₩23,100부터 시작됩니다.
지금 가입 >

OSLAB
Hallym University

Setup

- Notepad++
 - 초기 실행화면



■ 한 문장 출력하기

- “Hello Computer Architecture !” 라는 문구를 출력하는 프로그램
- 아래와 같은 프로그램을 Notepad 등으로 작성한 후 .asm 확장자로 저장한다.

```
.text
.globl main

main:
    li $v0, 4
    la $a0, str
    syscall

    li $v0, 10
    syscall

.data
str: .asciiz "Hello Computer Architecture !"
```

한 문장 출력하기

- QT-SPIM 초기 화면이다. 자신이 저장한 프로그램을 불러온 후 결과값 확인

```

User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c000000 jal 0x00000000 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)

```

```

User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)

```

```

[00400024] 34020004 ori $2, $0, 4 ; 6: li $v0, 4
[00400028] 3c011001 lui $1, 4097 [str] ; 7: la $a0, str
[0040002c] 34240000 ori $4, $1, 0 [str]
[00400030] 0000000c syscall ; 8: syscall
[00400034] 3402000a ori $2, $0, 10 ; 10: li $v0, 10
[00400038] 0000000c syscall ; 11: syscall

```

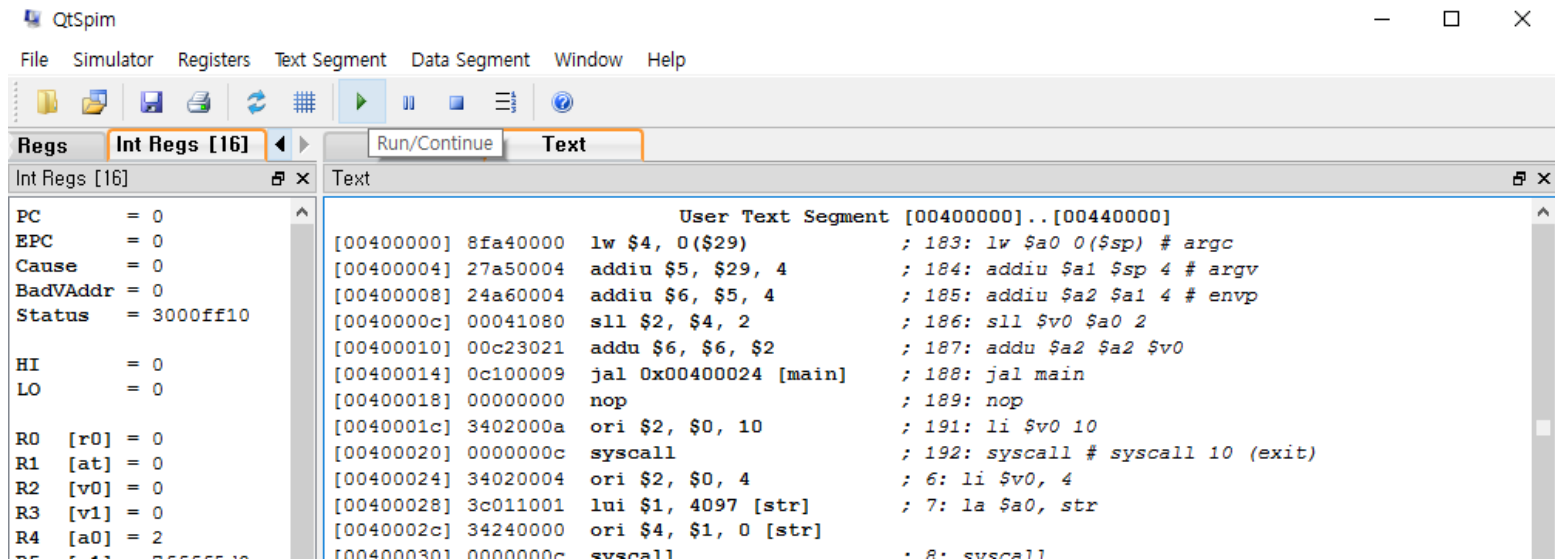
실제 작성한 프로그램은 여기서부터 시작함을 알수가 있다.

내가 작성한 코드를 확인할 수 있다.

LAB 0

한 문장 출력하기

- Run/Continue 버튼을 통해 컴파일을 수행




The image shows the QtSpim MIPS simulator interface. The 'Registers' tab is active, displaying the state of the MIPS registers. The 'Text' tab is also visible, showing the assembly code for the 'User Text Segment'.

Register	Value
PC	= 0
EPC	= 0
Cause	= 0
BadVAddr	= 0
Status	= 3000ff10
HI	= 0
LO	= 0
R0 [r0]	= 0
R1 [at]	= 0
R2 [v0]	= 0
R3 [v1]	= 0
R4 [a0]	= 2

The assembly code in the 'Text' tab is as follows:

```
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 34020004 ori $2, $0, 4 ; 6: li $v0, 4
[00400028] 3c011001 lui $1, 4097 [str] ; 7: la $a0, str
[0040002c] 34240000 ori $4, $1, 0 [str]
[00400030] 0000000c syscall ; 8: syscall
```

- Console 창을 통해 결과값 확인



The image shows the 'Console' window of the QtSpim simulator. It displays the output of the program, which is 'Hello Computer Architecture !'.

```
Hello Computer Architecture !
```

■ 분석

- 자신이 작성한 코드에 대해서 설명할 수 있어야 한다.

```
.text
.globl main

main:
    li $v0, 4
    la $a0, str
    syscall

    li $v0, 10
    syscall

.data
str: .asciiz "Hello Computer Architecture !"
```

- .text는 코드 세그먼트를 정의하는 것이다.
- .globl main은 main함수를 선언하는 것이다. 모든 MIPS 프로그램은 main에서 시작한다.
 - .globl은 다른 파일에서 reference 할 수 있도록 global symbol을 정의하는 것이다.
- li와 la는 pseudo instruction 중 하나이다.
 - li : load immediate, la : load address
 - 즉, li \$v0, 4는 \$v0 레지스터에 4번 코드를 불러와라 라는 뜻이 된다. 코드는 이후 다루겠다.
 - la \$a0, str 은 str의 주소를 \$a0 레지스터에 불러오라는 뜻이 된다.
- syscall은 \$v0 에 설정된 레지스터 값을 수행하게 된다.

■ 주석

- 주석은 개발자의 기본
- MIPS 환경에서는 # 을 넣어주면 이후 단어들은 주석처리가 된다.

■ ex)

```
.text
.globl main

main:           # 함수의 시작
    li $v0, 4
    la $a0, str
    syscall

    li $v0, 10
    syscall

.data
str: .asciiz "Hello Computer Architecture !"
```

■ 과제

- 실습한 내용의 화면캡처 본
- 실습 코드와 주석 (line by line)
- 위 두 가지를 워드 문서로 합하여 제출
- 수업시간 내 완료시 조교의 확인을 받고 퇴실 가능, 미확인시 결석처리

■ 파일명 ex) ca_00_학번_이름.docx

- 스마트 캠퍼스 과제란 제출 – 파일명 엄수

■ 제출기한

- 9월 7일 23:59까지