2022-1

Computer Architecture

MIPS Mars

Prof. Seong-Won Lee
광운대학교 컴퓨터 공학과

http://mpl.kw.ac.kr

JAVA Runtime Environment

- https://www.java.com/ko/

JAVA Runtime Environment

JAVA Runtime Environment

JAVA Runtime Environment

Mars

- http://courses.missouristate.edu/kenvollmar/mars/

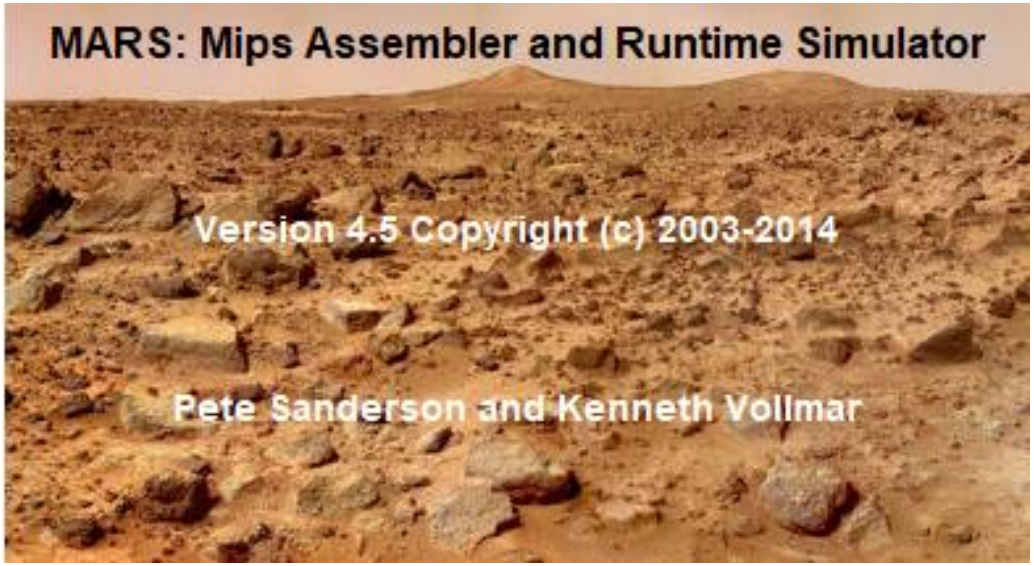
Click!

Mars


Click!
Missouri State UNIVERSITY
Search
a b c d e f g h i j k l m n o p q r s t u v w x y z
Download MARS
V4.5, Aug. 2014 (jar archive including Java source code)
Note: Is your MARS text unreadably small? Download and use a new release Java 9, which contains a fix to automatically scale and size AWT and Swing components for High Dots Per Inch (HiDPI) displays on Windows and Linux. Technical details.
Previous MARS version: MARS v4.4, Aug. 2013
Home
Features
Download
License
Papers
Help & Info
Contact Us
100% FREE
NO SPYWARE
NO ADWARE
NO VIRUSES
SOFTPEDIA
certified by www.softpedia.com
Feb. 2013: "MARS has been tested in the Softpedia labs using several industry-leading security solutions and found to be completely clean of adware/spyware components. ... Softpedia guarantees that MARS 4.3 is 100% FREE, which means it does not contain any form of malware, including spyware, viruses, trojans and backdoors."
Download MARS from Softpedia (version on Softpedia may lag behind the version on this page).
Option A: Desktop. Save the jar file on the desktop. Run MARS by double-clicking the icon.
Option B: DOS shell using jar file. Save the jar file in some folder. Open a DOS shell in that folder. Rename the jar file to "Mars.jar" for convenience. Run MARS with the DOS command java -jar Mars.jar
Option C: DOS shell using Java classes. Save the jar file in some folder. Open a DOS shell in that folder. Rename the jar file to "Mars.jar" for convenience. Extract MARS files with the DOS command jar -xf Mars.jar Run MARS with the DOS command java Mars
As of V4.0, MARS requires Java J2SE 1.5 (or later) SDK installed on your computer.
Download Java
Sample MIPS assembly program to run under MARS Fibonacci.asm
A "tool" is the MARS utility for MIPS control of simulated devices, including contention for resources. Sample tool is a Scavenger Hunt (screeenshot 23KB), in which each student writes a MIPS subroutine to direct the path of a character to travel to several locations. The companion Java "tool" program graphically displays the position of each character as the Scavenger Hunt game progresses. Interested instructors please contact Ken Vollmar for MIPS assembly solution or further information.
Missouri State University • Copyright 2006 Board of Governors, Missouri State University • Disclaimer • Accessibility
Maintained by: Ken Vollmar • Last Modified: December 8, 2017 • URL: http://courses.missouristate.edu/kenvollmar/mars/download.htm

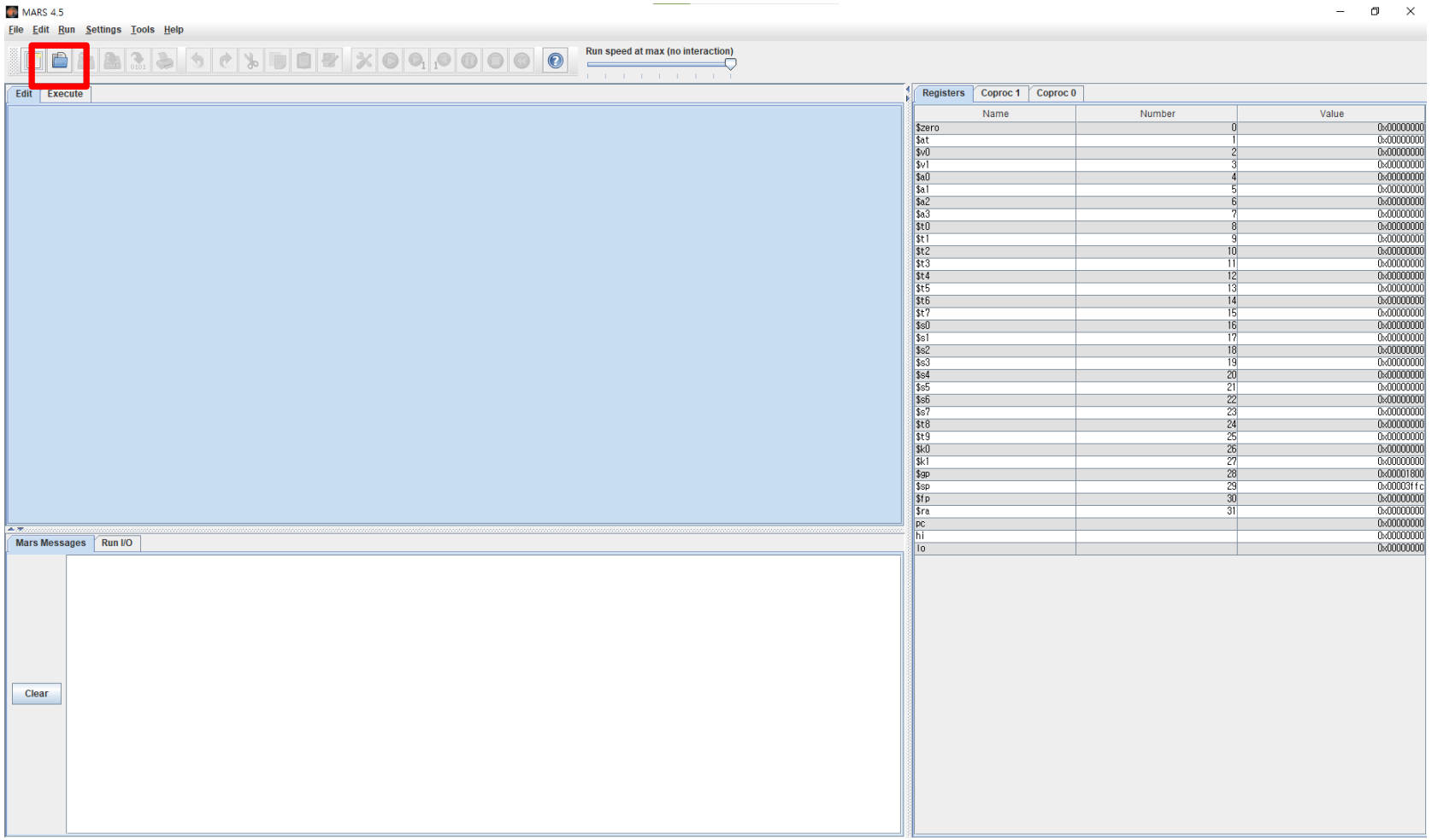
Mars

Mars4_5.jar

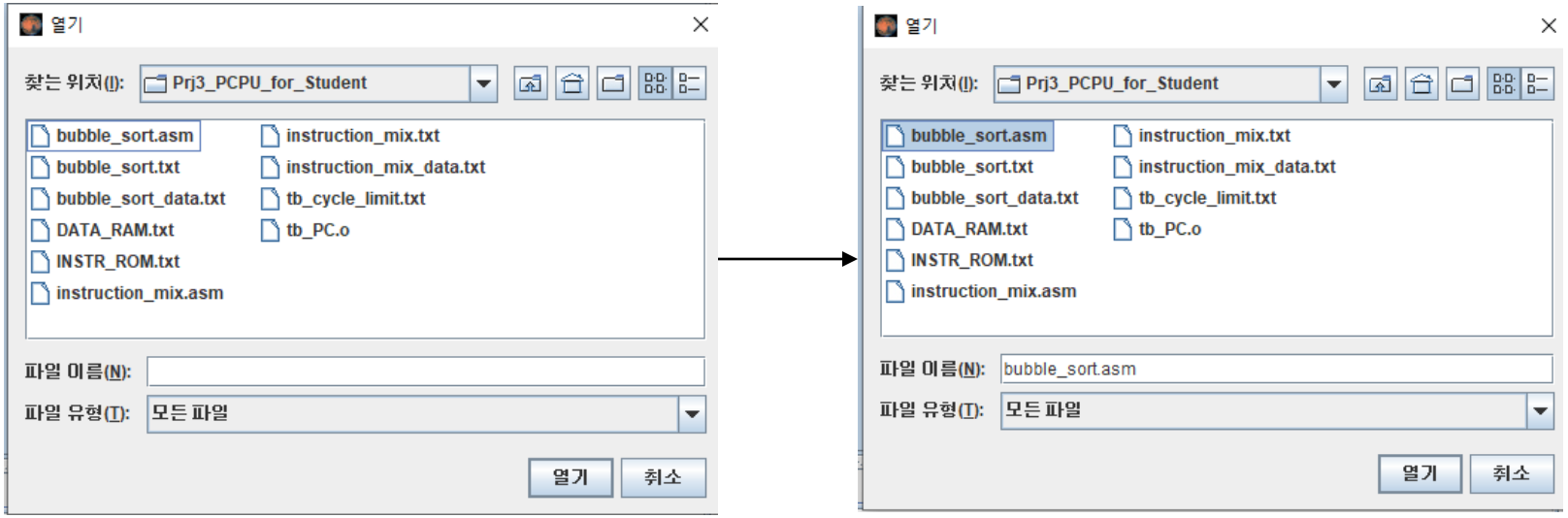


Mars4_5.jar
4.0/4.0MB

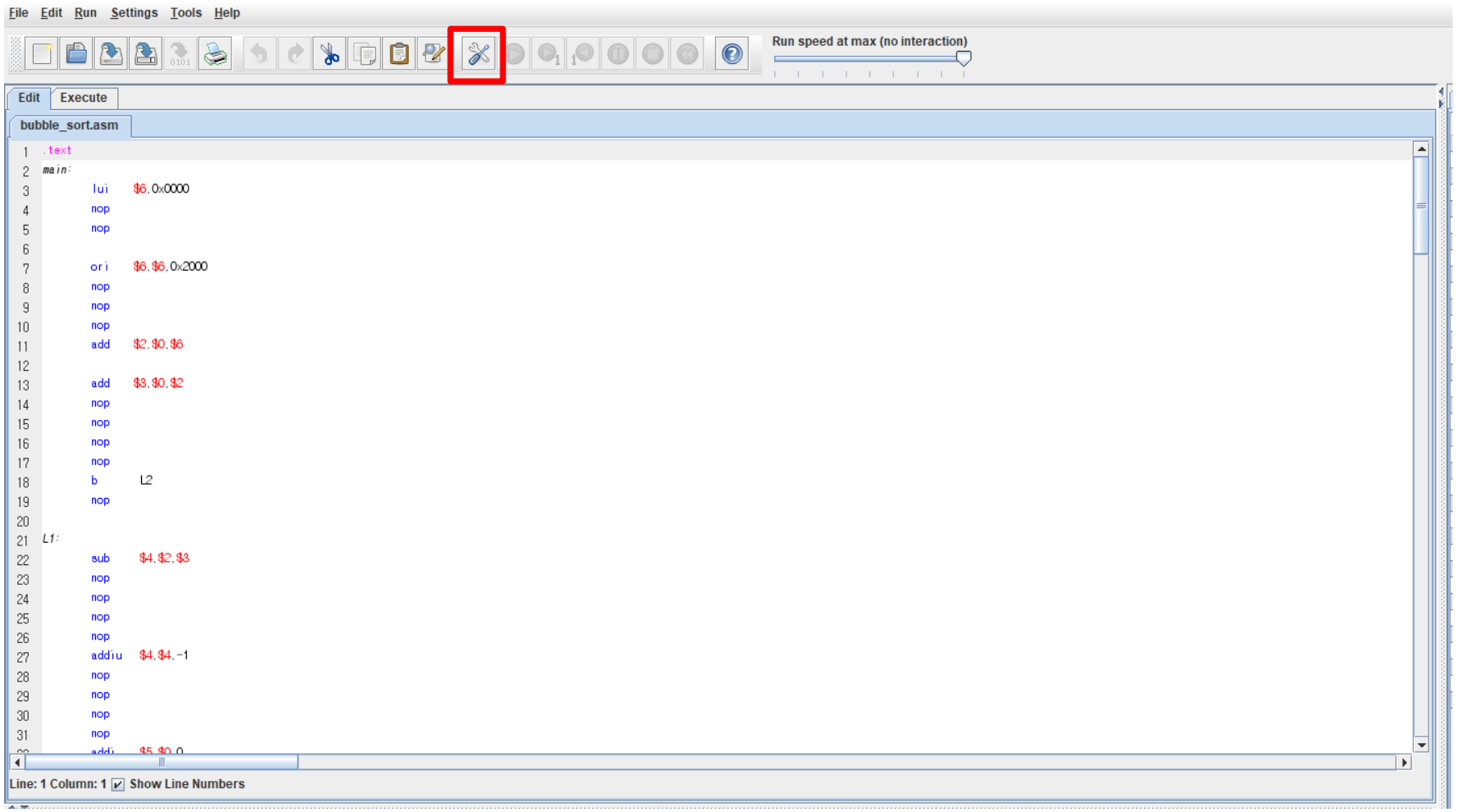
Mars


MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Mars Messages Run I/O
Clear
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000000
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000000
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x00001800
$sp 29 0x00003ffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00000000
hi 0x00000000
lo 0x00000000

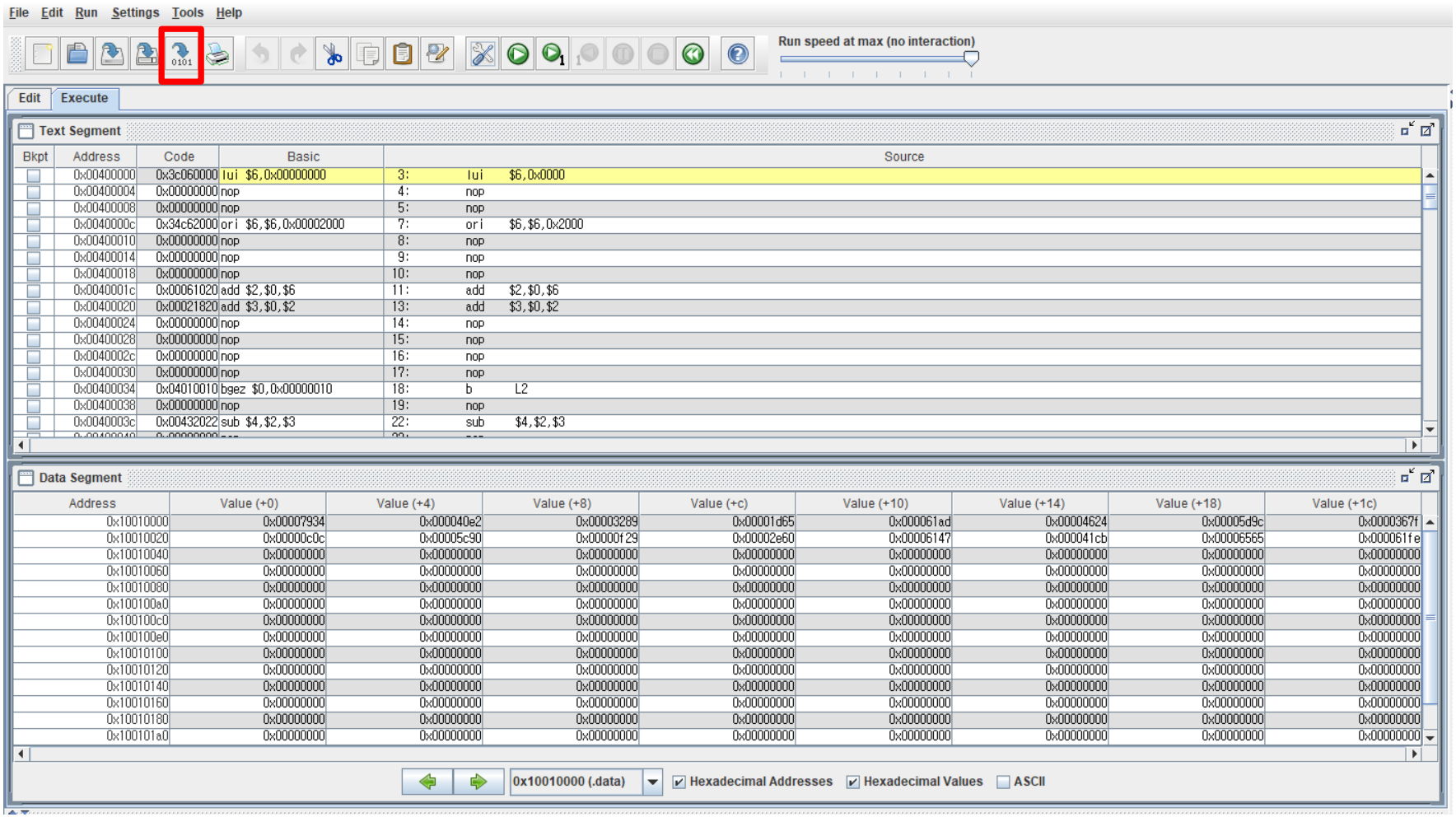
Mars


열기
찾는 위치(I): Prj3_PCPU_for_Student
bubble_sort.asm
bubble_sort.txt
bubble_sort_data.txt
DATA_RAM.txt
INSTR_ROM.txt
instruction_mix.asm
instruction_mix.txt
instruction_mix_data.txt
tb_cycle_limit.txt
tb_PC.o
파일 이름(N):
파일 유형(T): 모든 파일
열기
취소
열기
찾는 위치(I): Prj3_PCPU_for_Student
bubble_sort.asm
bubble_sort.txt
bubble_sort_data.txt
DATA_RAM.txt
INSTR_ROM.txt
instruction_mix.asm
instruction_mix.txt
instruction_mix_data.txt
tb_cycle_limit.txt
tb_PC.o
파일 이름(N): bubble_sort.asm
파일 유형(T): 모든 파일
열기
취소

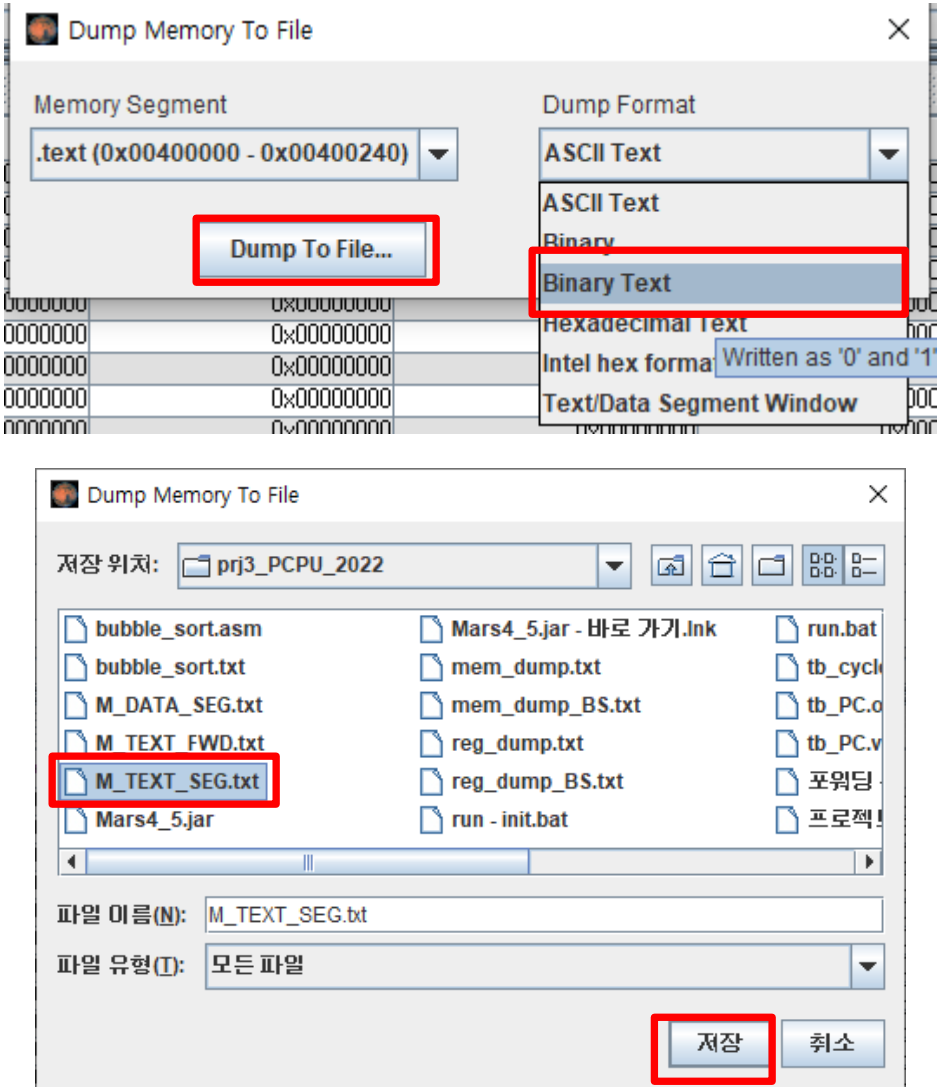
Mars


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
bubble_sort.asm
.text
main:
lui $6,0x0000
nop
nop
ori $6,$6,0x2000
nop
nop
nop
add $2,$0,$6
add $3,$0,$2
nop
nop
nop
nop
b L2
nop
L1:
sub $4,$2,$3
nop
nop
nop
nop
addiu $4,$4,-1
nop
nop
nop
nop
Line: 1 Column: 1 Show Line Numbers

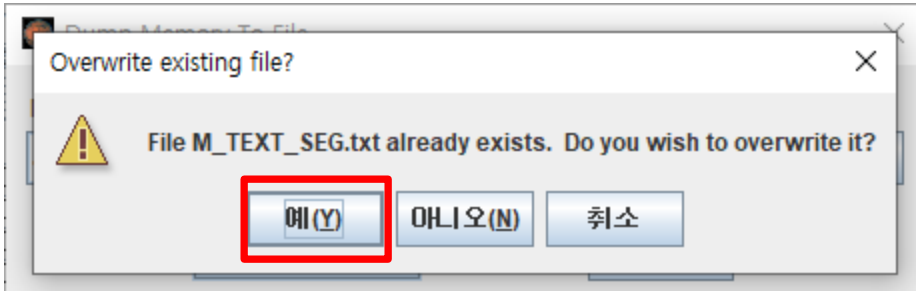
Mars

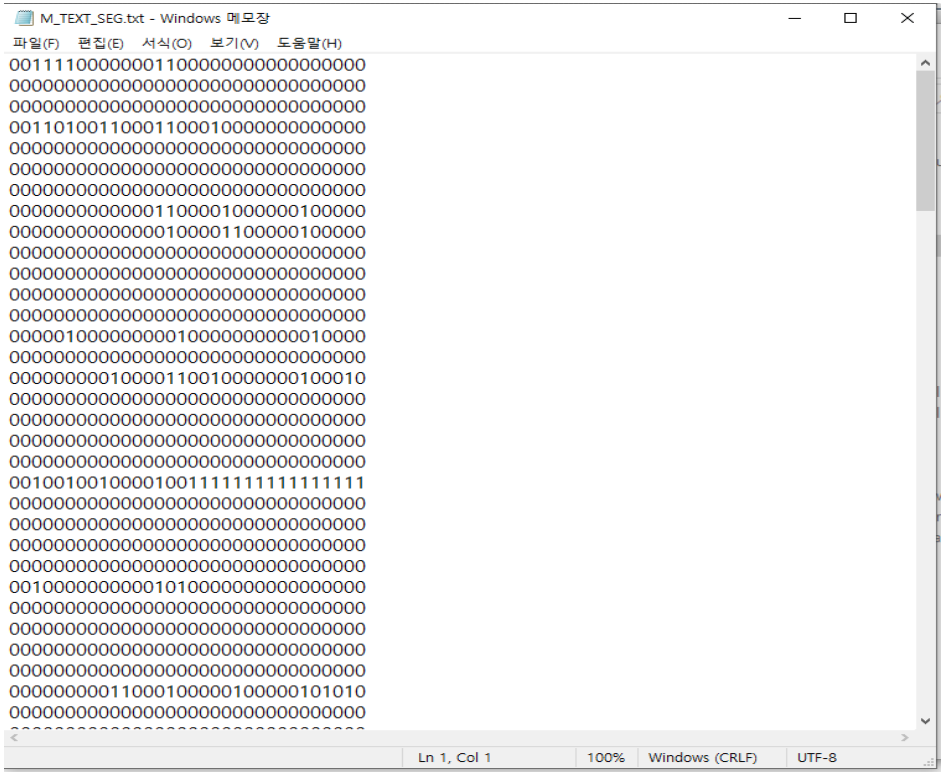

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x3c060000 lui $6,0x00000000 3: lui $6,0x0000
0x00400004 0x00000000 nop 4: nop
0x00400008 0x00000000 nop 5: nop
0x0040000c 0x34c62000 ori $6,$6,0x00002000 7: ori $6,$6,0x2000
0x00400010 0x00000000 nop 8: nop
0x00400014 0x00000000 nop 9: nop
0x00400018 0x00000000 nop 10: nop
0x0040001c 0x00061020 add $2,$0,$6 11: add $2,$0,$6
0x00400020 0x00021820 add $3,$0,$2 13: add $3,$0,$2
0x00400024 0x00000000 nop 14: nop
0x00400028 0x00000000 nop 15: nop
0x0040002c 0x00000000 nop 16: nop
0x00400030 0x00000000 nop 17: nop
0x00400034 0x04010010 bgez $0,0x00000010 18: b L2
0x00400038 0x00000000 nop 19: nop
0x0040003c 0x00432022 sub $4,$2,$3 22: sub $4,$2,$3
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00007934 0x000040e2 0x00003289 0x00001d65 0x000061ad 0x00004624 0x00005d9c 0x0000367f
0x10010020 0x00000c0c 0x00005c90 0x00000f29 0x00002e60 0x00006147 0x000041cb 0x00006565 0x000061fe
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100e0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010100 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010120 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010140 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010160 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010180 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100101a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII

Mars

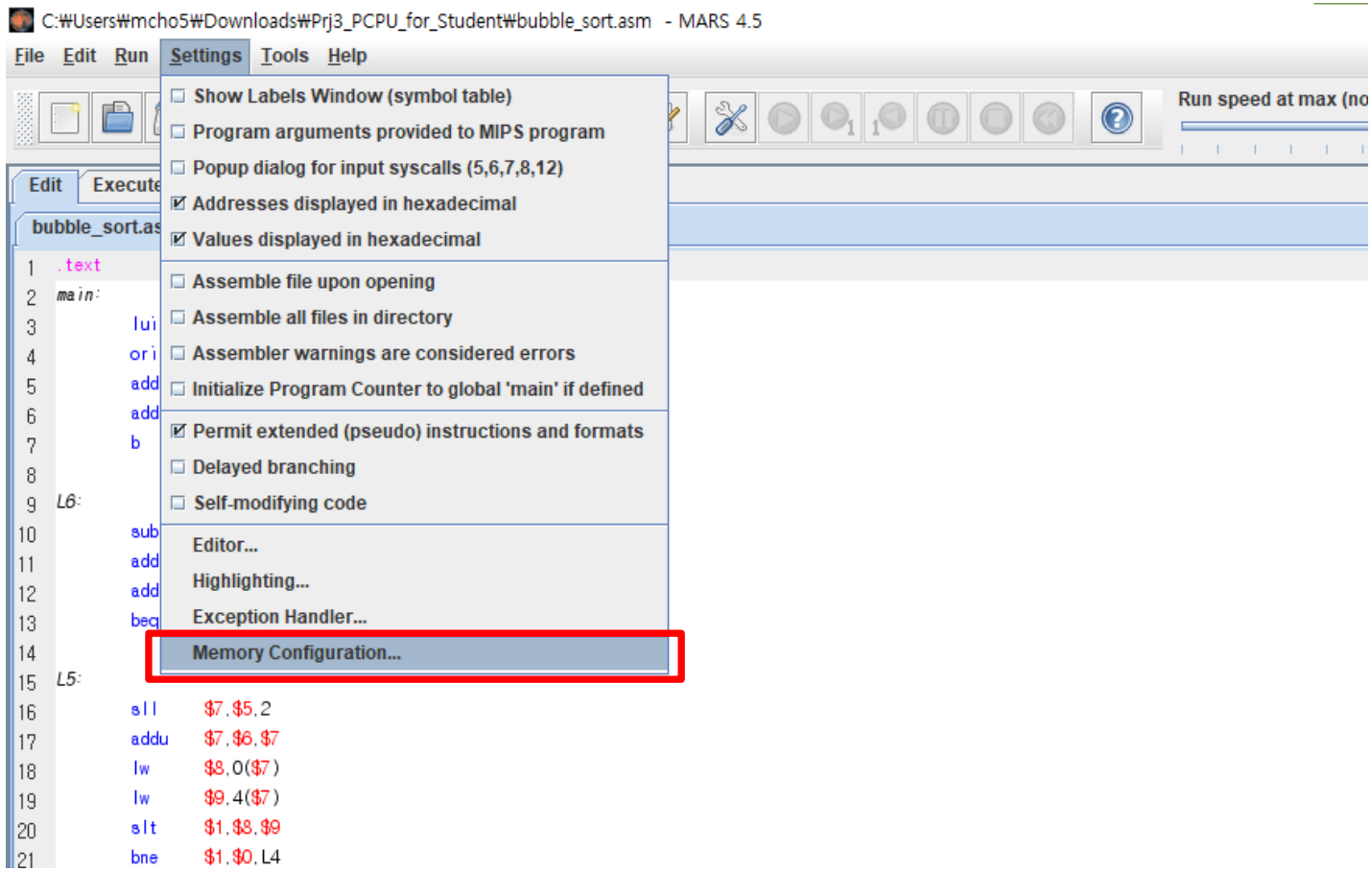

Dump Memory To File
Memory Segment
.text (0x00400000 - 0x00400240)
Dump Format
ASCII Text
Dump To File...
ASCII Text
Binary
Binary Text
Hexadecimal Text
Intel hex forma
Written as '0' and '1'
Text/Data Segment Window
0x00000000
Dump Memory To File
저장 위치:
prj3_PCPU_2022
bubble_sort.asm
bubble_sort.txt
M_DATA_SEG.txt
M_TEXT_FWD.txt
M_TEXT_SEG.txt
Mars4_5.jar
Mars4_5.jar - 바로 가기.lnk
mem_dump.txt
mem_dump_BS.txt
reg_dump.txt
reg_dump_BS.txt
run - init.bat
run.bat
파일 이름(N):
M_TEXT_SEG.txt
파일 유형(T):
모든 파일
저장
취소

Mars

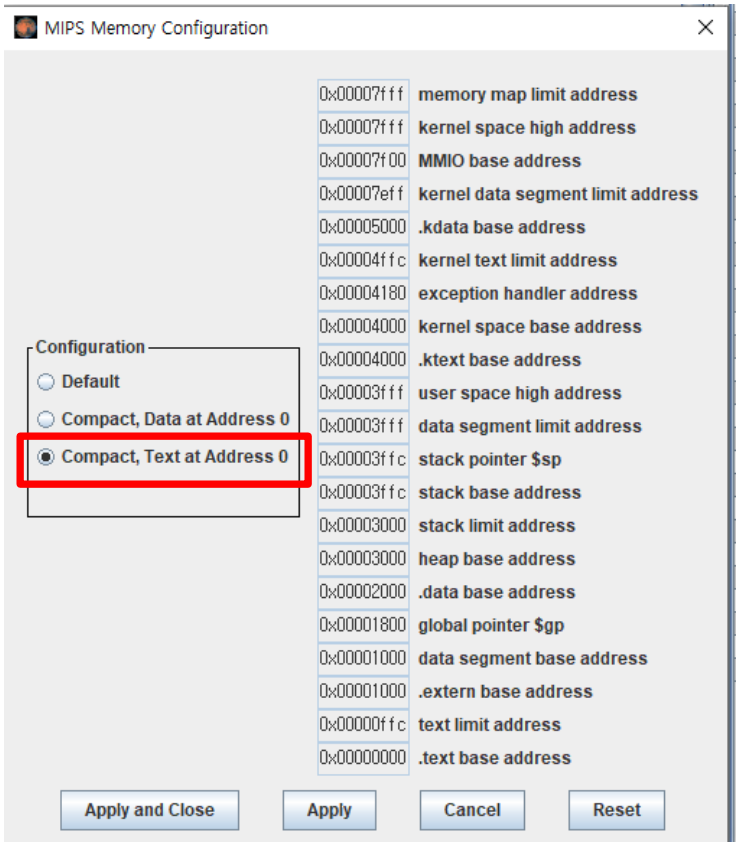

Overwrite existing file?
File M_TEXT_SEG.txt already exists. Do you wish to overwrite it?
예(Y)
아니오(N)
취소


M_TEXT_SEG.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
00111100000001100000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00110100110001100010000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000110000100000100000
00000000000000010000110000100000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00001000000000100000000000010000
00000000000000000000000000000000
00000000010000110010000000100010
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00100100100001001111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00100000000010100000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000011000100000100000101010
00000000000000000000000000000000
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

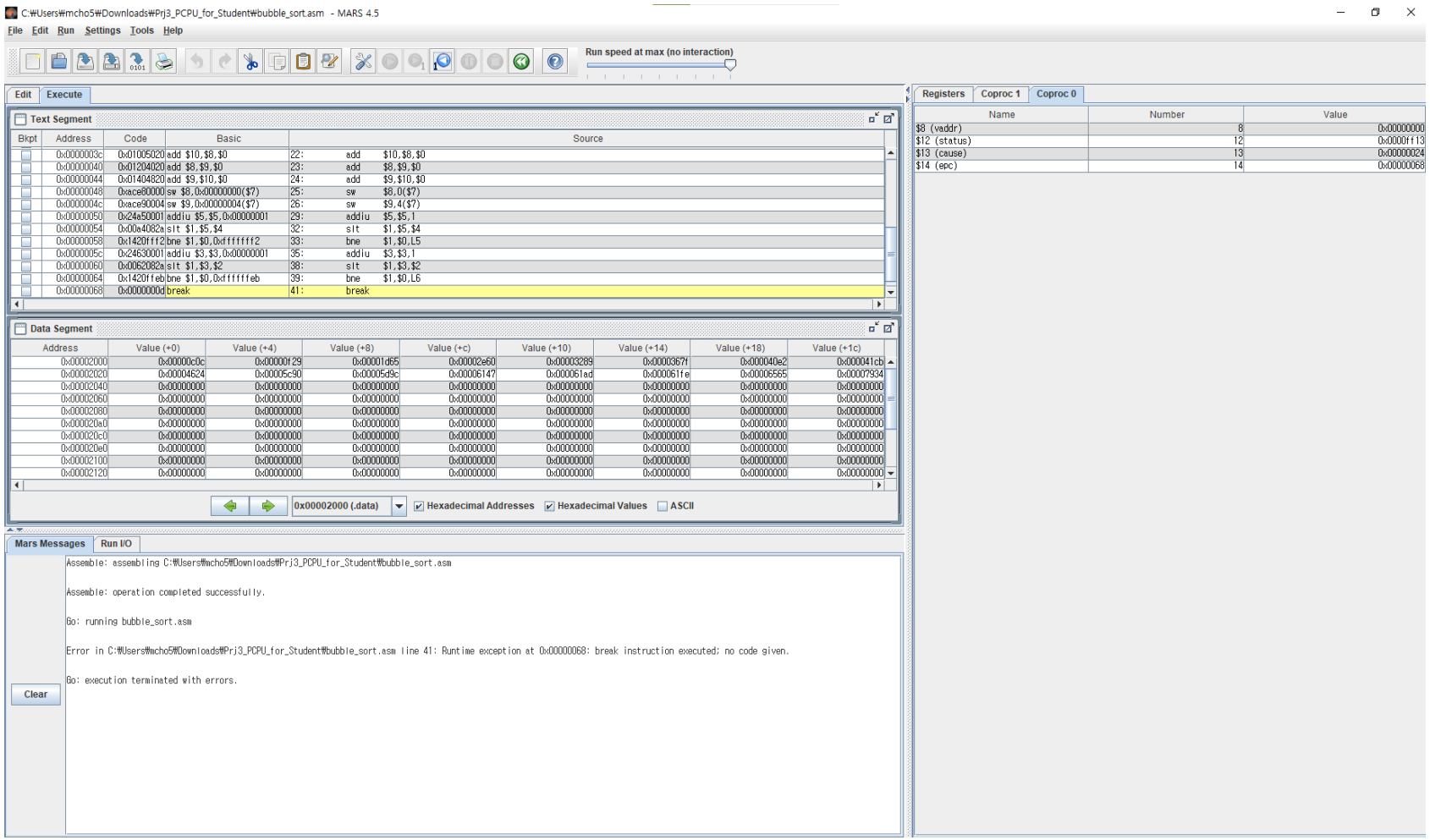
Mars


C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no
Edit Execute
bubble_sort.as
Show Labels Window (symbol table)
Program arguments provided to MIPS program
Popup dialog for input syscalls (5,6,7,8,12)
Addresses displayed in hexadecimal
Values displayed in hexadecimal
Assemble file upon opening
Assemble all files in directory
Assembler warnings are considered errors
Initialize Program Counter to global 'main' if defined
Permit extended (pseudo) instructions and formats
Delayed branching
Self-modifying code
Editor...
Highlighting...
Exception Handler...
Memory Configuration...
1 .text
2 main:
9 L6:
15 L5:
16 sll $7,$5,2
17 addu $7,$6,$7
18 lw $8,0($7)
19 lw $9,4($7)
20 slt $1,$8,$9
21 bne $1,$0,L4

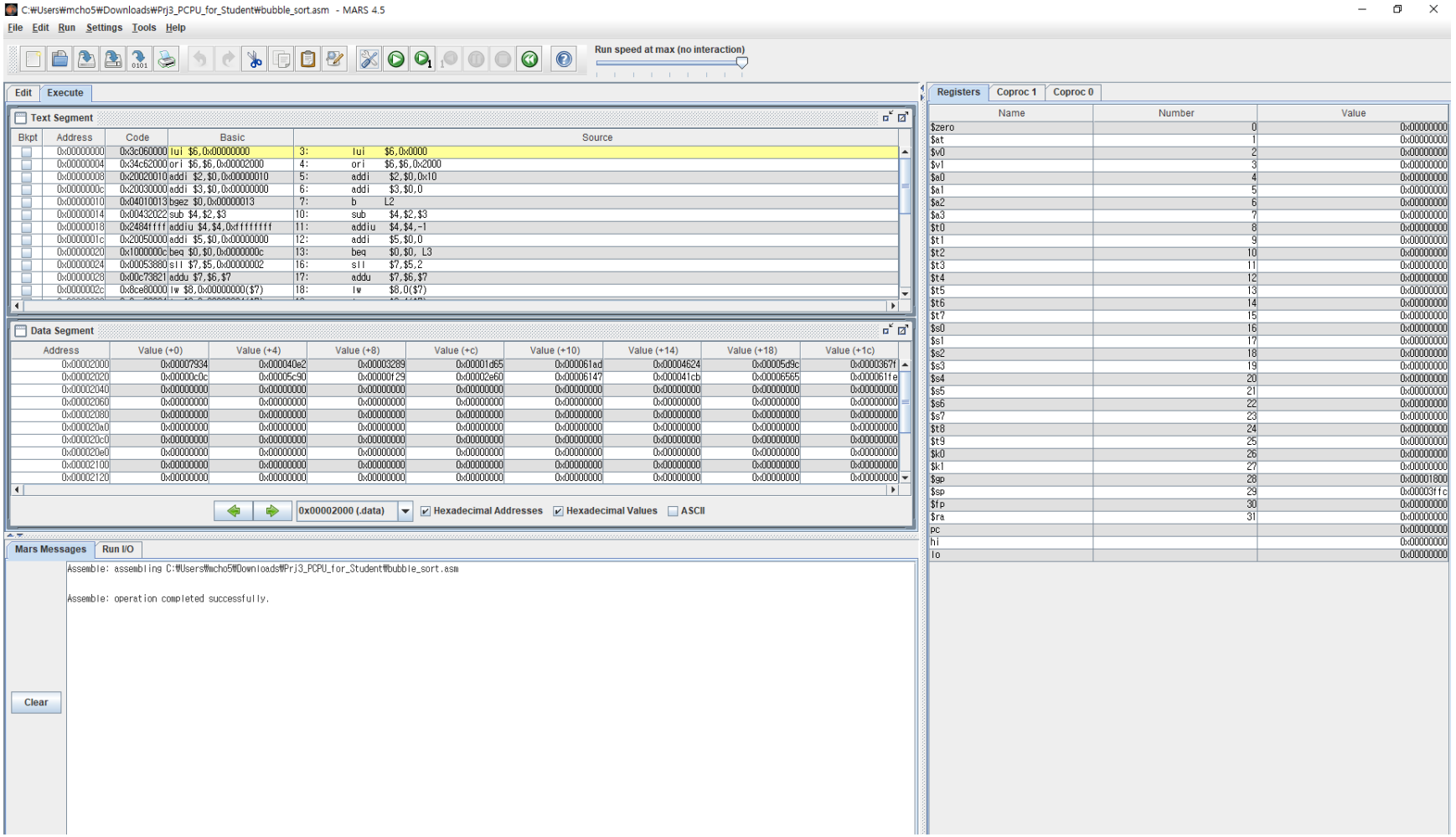
Mars


MIPS Memory Configuration
Configuration
Default
Compact, Data at Address 0
Compact, Text at Address 0
0x00007fff memory map limit address
0x00007fff kernel space high address
0x00007f00 MMIO base address
0x00007eff kernel data segment limit address
0x00005000 .kdata base address
0x00004ffc kernel text limit address
0x00004180 exception handler address
0x00004000 kernel space base address
0x00004000 .ktext base address
0x00003fff user space high address
0x00003fff data segment limit address
0x00003ffc stack pointer $sp
0x00003ffc stack base address
0x00003000 stack limit address
0x00003000 heap base address
0x00002000 .data base address
0x00001800 global pointer $gp
0x00001000 data segment base address
0x00001000 .extern base address
0x00000ffc text limit address
0x00000000 .text base address
Apply and Close
Apply
Cancel
Reset

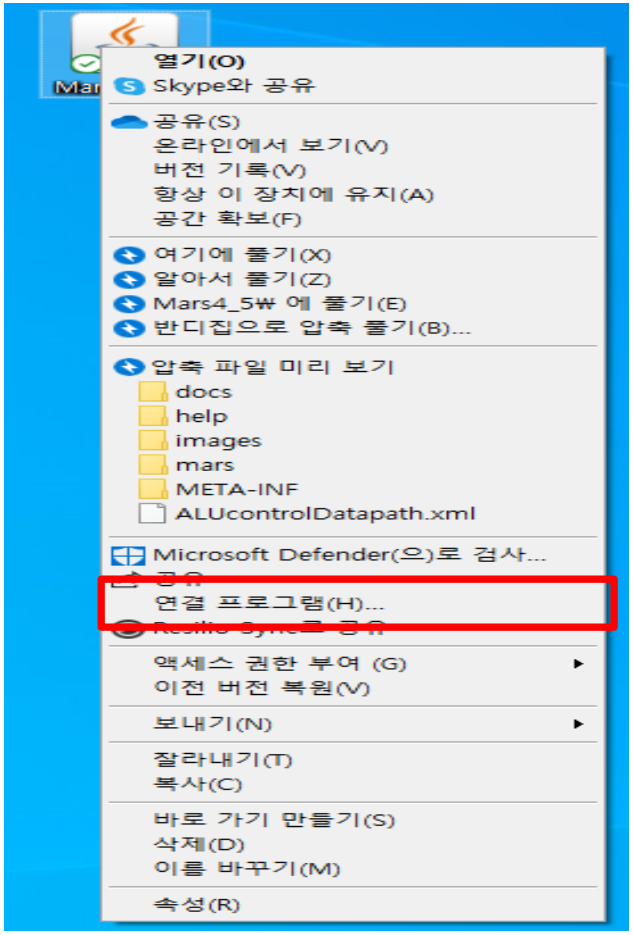
Mars

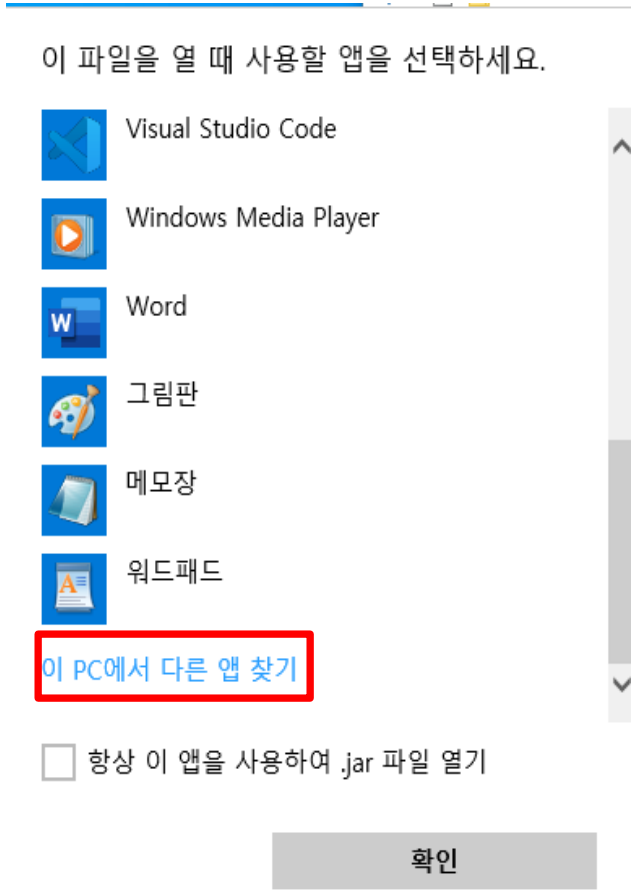

C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x0000003c 0x01005020 add $10,$8,$0 22: add $10,$8,$0
0x00000040 0x01204020 add $8,$9,$0 23: add $8,$9,$0
0x00000044 0x01404820 add $9,$10,$0 24: add $9,$10,$0
0x00000048 0xace80000 sw $8,0x00000000($7) 25: sw $8,0($7)
0x0000004c 0xace90004 sw $9,0x00000004($7) 26: sw $9,4($7)
0x00000050 0x24a50001 addiu $5,$5,0x00000001 29: addiu $5,$5,1
0x00000054 0x00a4082a slt $1,$5,$4 32: slt $1,$5,$4
0x00000058 0x1420fff2 bne $1,$0,0xfffffff2 33: bne $1,$0,L5
0x0000005c 0x24630001 addiu $3,$3,0x00000001 35: addiu $3,$3,1
0x00000060 0x0062082a slt $1,$3,$2 38: slt $1,$3,$2
0x00000064 0x1420ffeb bne $1,$0,0xffffffeb 39: bne $1,$0,L6
0x00000068 0x0000000d break 41: break
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x00002000 0x00000c0c 0x00000f29 0x00001d65 0x00002e60 0x00003289 0x0000367f 0x000040e2 0x000041cb
0x00002020 0x00004624 0x00005c90 0x00005d9c 0x00006147 0x000061ad 0x00006ffe 0x00006565 0x00007934
0x00002040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020e0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002100 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002120 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages Run I/O
Assemble: assembling C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm
Assemble: operation completed successfully.
Go: running bubble_sort.asm
Error in C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm line 41: Runtime exception at 0x00000068: break instruction executed; no code given.
Go: execution terminated with errors.
Clear
Registers Coproc 1 Coproc 0
Name Number Value
$8 (vaddr) 8 0x00000000
$12 (status) 12 0x0000ff13
$13 (cause) 13 0x00000024
$14 (epc) 14 0x00000068

Mars

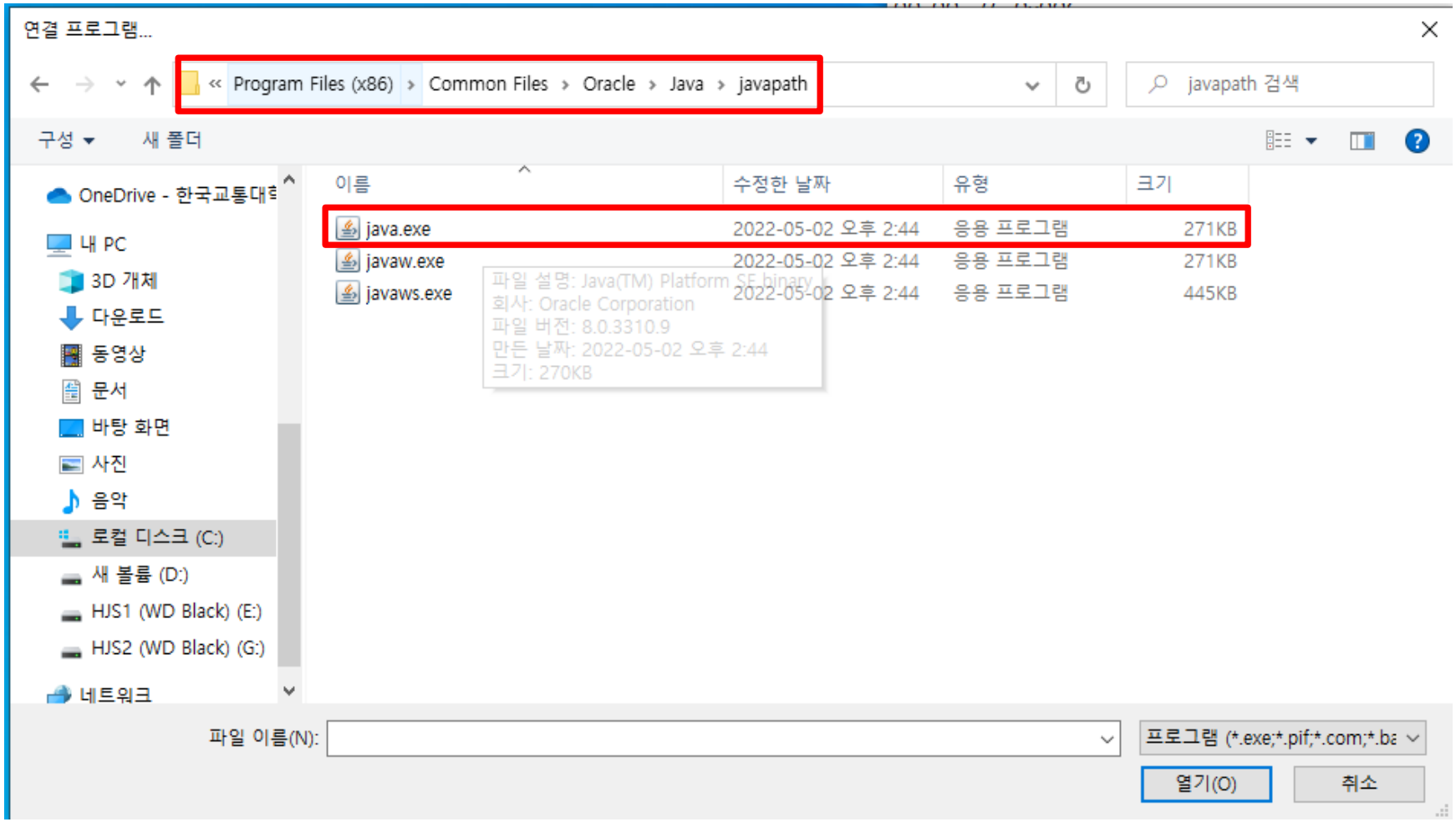

C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00000000 0x3c060000 lui $6,0x00000000 3: lui $6,0x0000
0x00000004 0x34c62000 ori $6,$6,0x00002000 4: ori $6,$6,0x2000
0x00000008 0x20020010 addi $2,$0,0x00000010 5: addi $2,$0,0x10
0x0000000c 0x20030000 addi $3,$0,0x00000000 6: addi $3,$0,0
0x00000010 0x04010013 bgez $0,0x00000013 7: b L2
0x00000014 0x00432022 sub $4,$2,$3 10: sub $4,$2,$3
0x00000018 0x2484ffff addiu $4,$4,0xffffffff 11: addiu $4,$4,-1
0x0000001c 0x20050000 addi $5,$0,0x00000000 12: addi $5,$0,0
0x00000020 0x1000000c beq $0,$0,0x0000000c 13: beq $0,$0, L3
0x00000024 0x00053880 sll $7,$5,0x00000002 16: sll $7,$5,2
0x00000028 0x00c73821 addu $7,$6,$7 17: addu $7,$6,$7
0x0000002c 0x8ce80000 lw $8,0x00000000($7) 18: lw $8,0($7)
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x00002000 0x00007934 0x000040e2 0x00003289 0x00001d65 0x000061ad 0x00004624 0x00005d9c 0x0000367f
0x00002020 0x00000c0c 0x00005c90 0x00000f29 0x00002e60 0x00006147 0x000041cb 0x00006565 0x00001fe
0x00002040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x000020e0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002100 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002120 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00002000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages Run I/O
Assemble: assembling C:₩Users₩mcho5₩Downloads₩Prj3_PCPU_for_Student₩bubble_sort.asm
Assemble: operation completed successfully.
Clear
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x00000000
$t1 9 0x00000000
$t2 10 0x00000000
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000000
$s1 17 0x00000000
$s2 18 0x00000000
$s3 19 0x00000000
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x00001800
$sp 29 0x00003ffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00000000
hi 0x00000000
lo 0x00000000

Mars 바로가기 만드는 법

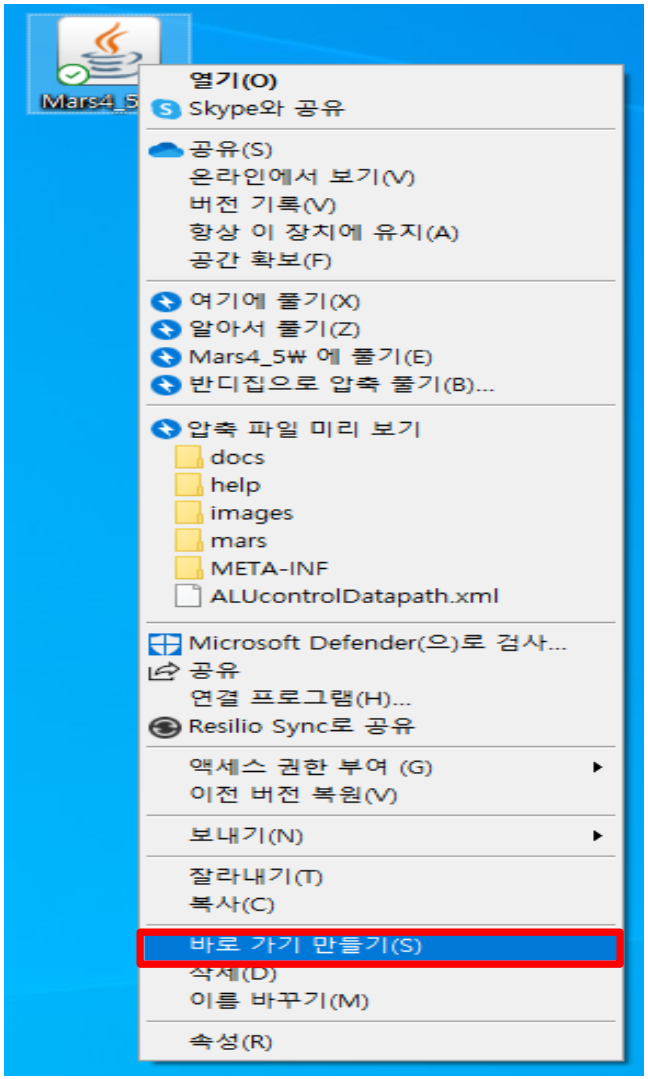


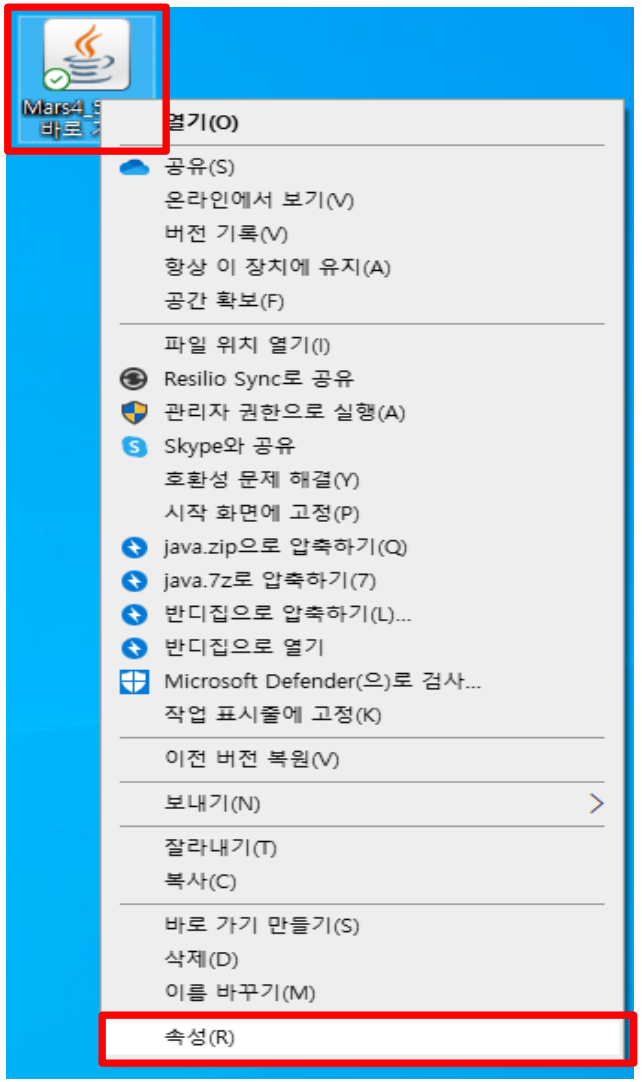


Mars 바로가기 만드는 법



Mars 바로가기 만드는 법





Mars 바로가기 만드는 법