

Computer Architecture

HW#2 Single-cycle MIPS

Dept. of Computer Engineering

Jaehyun Nam

jaehyun.nam@dankook.ac.kr

Single-cycle MIPS Emulator

- Make a MIPS CPU emulator
- Execute MIPS Instructions (except for floating point operations)
- Assume that you have memory and a program, which is loaded in memory before execution
- Take MIPS a binary program
- Display the state changes at every clock cycle
- Handle exception gracefully

Green Sheet

- https://inst.eecs.berkeley.edu/~cs61c/resources/MIPS_Green_Sheet.pdf

③

OPCODES, BASE CONVERSION, ASCII SYMBOLS

MIPS opcode (31:26)	(1) MIPS funct (5:0)	(2) MIPS funct (5:0)	Binary	Deci- mal	Hexa- deci- mal	ASCII Char- acter	Deci- mal	Hexa- deci- mal	ASCII Char- acter
(1)	sll	add.f	00 0000	0	0	NUL	64	40	@
		sub.f	00 0001	1	1	SOH	65	41	A
j	srl	mul.f	00 0010	2	2	STX	66	42	B
jal	sra	div.f	00 0011	3	3	ETX	67	43	C
beq	sllv	sqr.f	00 0100	4	4	EOT	68	44	D
bne		abs.f	00 0101	5	5	ENQ	69	45	E
blez	srlv	mov.f	00 0110	6	6	ACK	70	46	F
bgtz	srav	neg.f	00 0111	7	7	BEL	71	47	G
addi	jr		00 1000	8	8	BS	72	48	H
addiu	jalr		00 1001	9	9	HT	73	49	I
slti	movz		00 1010	10	a	LF	74	4a	J
sltiu	movn		00 1011	11	b	VT	75	4b	K
andi	syscall	round.w.f	00 1100	12	c	FF	76	4c	L
ori	break	trunc.w.f	00 1101	13	d	CR	77	4d	M

Requirements

- Before the execution, the binary file is loaded into the memory
 - Note that the memory can be a data structure defined with a large array
- Read all the file content into your memory (data structure)
- Assume that all register values are all zero, except for
 - RA(r31) and SP(r29), of which value is 0xFFFFFFFF and 0x1000000
- When your PC becomes 0xFFFFFFFF, your machine completes execution
- Your application is loaded to 0x0, and stack pointer is 0x1000000
- Need to fix the jump address of each function call in manual

Requirements

- MIPS executes instructions in the following stages:
 - Instruction fetch
 - Instruction is moved from memory to CPU
 - Instruction decode
 - Instruction is decoded
 - Execution
 - ALU operates, and the calculation result is out
 - Load/store result to memory
 - load/store memory operation is completed
 - Write back to reg. file
 - Update register values (including PC)

Requirements

- Single-cycle MPS processor performs all five stages in the same cycle
- The emulator must implement functions that correspond to each stage
- To begin the machine, it moves PC value to the very beginning point of the program, which is the address zero
- At the end of each cycle, the emulator prints out all of the changed states from the previous ones
 - A set of general registers, PC, and memory
 - Print out only changed states

Programming Language

- **C** is highly recommended since we need **C** for the following assignments
- But **Java** and **Python** are also okay in this assignment

Document

- Introduction
 - Brief description of the assignment
- Background
 - Important concept, specific considerations for implementation
- Implementation
 - How you organized your program (**design**)
 - What parts you implemented and what parts you didn't (couldn't)
 - Including the implementation for extra points
 - Explain why
- **Environment**
 - **How to build** the development **environment** for testing (+ screenshots)
 - Should be specific
 - If I can't create an environment by doing what you've described, I won't grade your code
 - **How to compile and run** your program
 - Should be specific
 - **Screenshots** working proofs with explanation
- Lesson
 - What was hard, What you thought while doing the assignment, etc.

E-mail Submission

- Send your assignment to jaehyun.nam@dankook.ac.kr
- E-mail title
 - [2023-1 ARCHITECTURE] HW2 [StudentID] [Name]
 - Ex) [2023-1 ARCHITECTURE] HW2 12345678 남재현
- E-mail body
 - **YOU MUST CHANGE THE NAMES OF YOUR CODE AND DOCUMENT AS FOLLOWS**
 - **IF NOT, YOU WILL GET A PENALTY**
 - Source code
 - **HW2_StudentID_Name_code.zip**
 - Ex) [HW2_12345678_남재현_code.zip](#)
 - Document
 - **HW2_StudentID_Name_document.pdf**
 - Ex) [HW2_12345678_남재현_document.pdf](#)

Due Date

- 2:30PM on June 2nd, 2023
 - Right before class

Appendix

Dept. of Computer Engineering

Jaehyun Nam

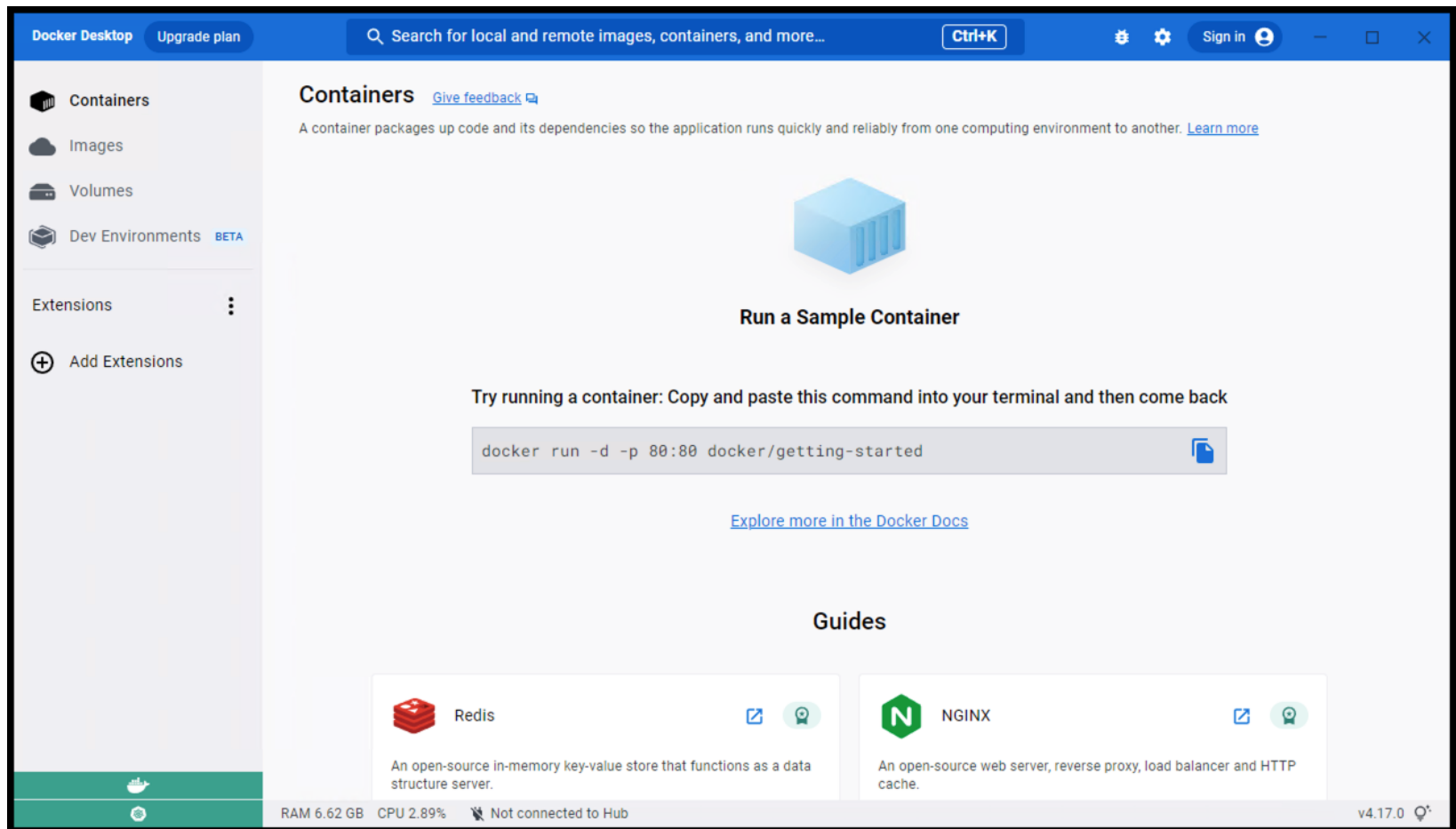
jaehyun.nam@dankook.ac.kr

MIPS Binary

- To make MIPS binary, you need to use MIPS cross-compiler toolchain
 - `sudo apt-get update`
 - `sudo apt-get -y install gcc-mips-linux-gnu`
- First, write C code (`input.c`)
- Compile the code with `mips-gcc` (compile only)
 - `mips-linux-gnu-gcc -c input.c -mips1 -mfp32`
- Translate the object binary, stripping ELF headers
 - `mips-linux-gnu-objcopy -O binary -j .text input.o input.bin`
 - `mips-linux-gnu-objdump -d input.o`
 - `hexdump input.bin`

Docker Environment for MIPS Compilation

- Docker Desktop
 - <https://www.docker.com/products/docker-desktop/>



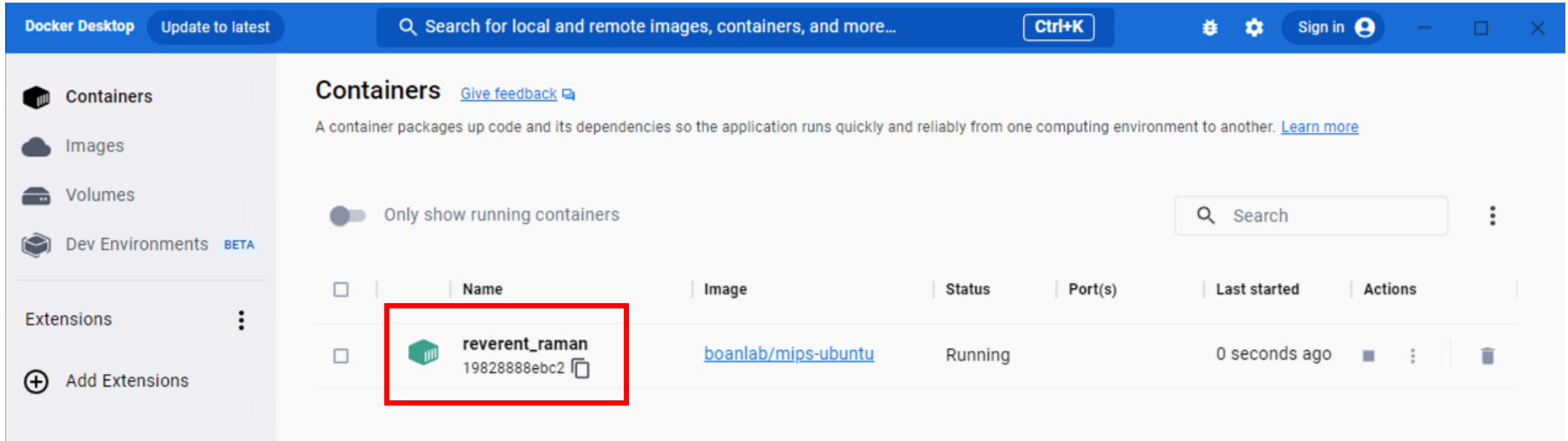
Docker Environment for MIPS Compilation

- PowerShell (or some other terminal)
 - `docker run -it --rm boanlab/mips-ubuntu`

```
PS C:\Users\j\j> docker run -it --rm boanlab/mips-ubuntu
Unable to find image 'boanlab/mips-ubuntu:latest' locally
latest: Pulling from boanlab/mips-ubuntu
dbf6a9befcde: Pull complete
2f8028e30bc0: Pull complete
e7577e005956: Pull complete
Digest: sha256:9e781aad0a0a9695aadcb28817aab5fcae5b8e0f57d150e89f42e8deaa958a0d
Status: Downloaded newer image for boanlab/mips-ubuntu:latest
root@fb24150baa75:/#
```

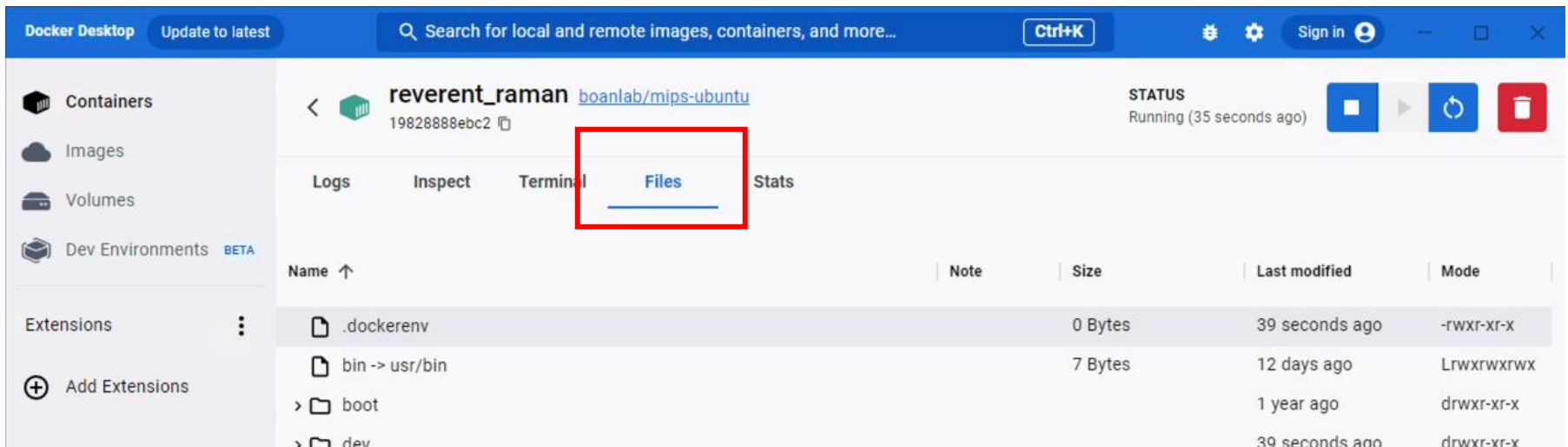
Docker Environment for MIPS Compilation

- 생성된 컨테이너 선택



The screenshot shows the Docker Desktop interface. On the left sidebar, the 'Containers' tab is selected. The main area displays a table of containers. One container, 'reverent_raman', is highlighted with a red box. The container's details are as follows:

Name	Image	Status	Port(s)	Last started	Actions
reverent_raman 1982888ebc2	boanlab/mips-ubuntu	Running		0 seconds ago	[Stop] [Refresh] [Delete]



The screenshot shows the Docker Desktop interface with the 'reverent_raman' container selected. The 'Files' tab is highlighted with a red box. The container's details are as follows:

reverent_raman boanlab/mips-ubuntu
1982888ebc2

STATUS: Running (35 seconds ago)

Logs Inspect Terminal **Files** Stats

Name	Note	Size	Last modified	Mode
.dockerenv		0 Bytes	39 seconds ago	-rwxr-xr-x
bin -> usr/bin		7 Bytes	12 days ago	Lrwxrwxrwx
boot			1 year ago	drwxr-xr-x
dev			39 seconds ago	drwxr-xr-x

Docker Environment for MIPS Compilation

The screenshot shows the Docker Desktop interface. On the left sidebar, the 'Containers' section is active. The main panel displays the details of a container named 'reverent_raman' (ID: 19828888ebc2), which is running the image 'boanlab/mips-ubuntu'. The 'Files' tab is selected, showing a list of files and directories within the container. A red box highlights the 'root' directory. The status bar at the bottom indicates the host has 6.79 GB of RAM and 2.51% CPU usage, and is not connected to the Docker Hub.

Name ↑	Note	Size	Last modified	Mode
lib64 -> usr/lib64		9 Bytes	12 days ago	Lrwxrwxrwx
libx32 -> usr/libx32		10 Bytes	12 days ago	Lrwxrwxrwx
> media			12 days ago	drwxr-xr-x
> mnt			12 days ago	drwxr-xr-x
> opt			12 days ago	drwxr-xr-x
> proc			1 minute ago	dr-xr-xr-x
✓ > root			12 days ago	drwx-----
.bashrc		3 kB	2 years ago	-rw-r--r--
.profile		161 Bytes	4 years ago	-rw-r--r--
> run			12 days ago	drwxr-xr-x
sbin -> usr/sbin		8 Bytes	12 days ago	Lrwxrwxrwx
> srv			12 days ago	drwxr-xr-x
> sys			1 minute ago	dr-xr-xr-x
> tmp			10 minutes ago	dtrwxrwxr...

RAM 6.79 GB CPU 2.51% Not connected to Hub v4.17.0

Docker Environment for MIPS Compilation

The screenshot shows the Docker Desktop interface. On the left sidebar, the 'Containers' section is active, showing a container named 'reverent_raman' with ID '1982888ebc2'. The container is running. The 'Files' tab is selected, displaying a file tree. The 'root' directory is highlighted with a red box. A file explorer window is overlaid on the container's file view, showing a file named 'input' being dragged from the local file system to the 'root' directory. A red arrow points from the 'input' file in the file explorer to the 'root' directory in the container's file view, with the text 'Drag and Drop' written in red. The file explorer window shows the path 'C:\Users\Hjn\sample_data' and the file 'input' is selected. The container's file view shows the 'root' directory with files like 'lib64', 'libx32', 'media', 'mnt', 'opt', 'proc', 'run', 'sbin', 'srv', and 'sys'. The 'root' directory is highlighted with a red box. The file explorer window shows the file 'input' being dragged. A red arrow points from the 'input' file in the file explorer to the 'root' directory in the container's file view, with the text 'Drag and Drop' written in red.

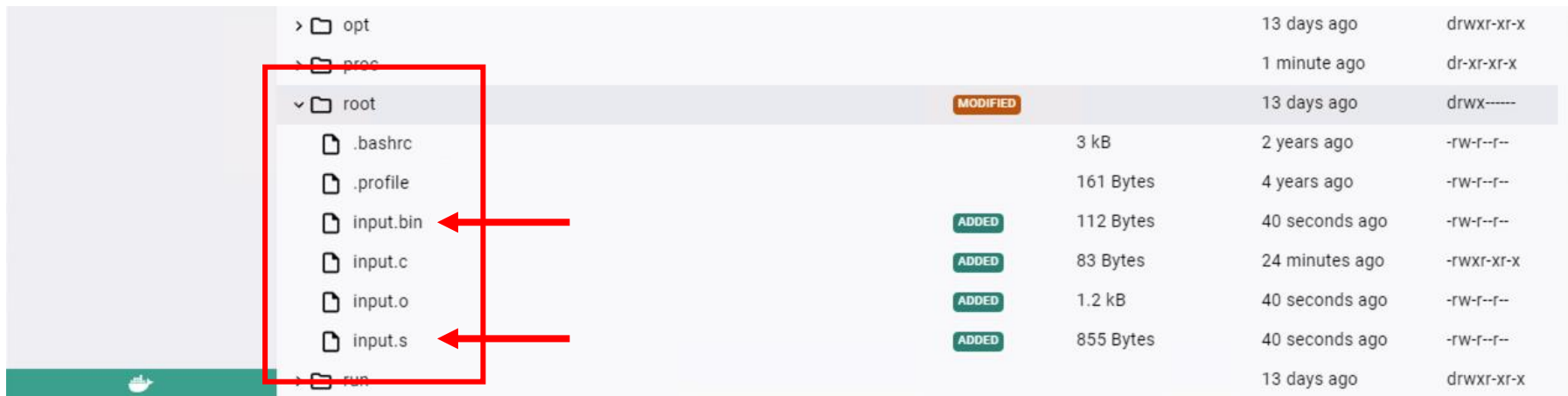
Docker Desktop interface showing a container named **reverent_raman** (ID: 1982888ebc2) running. The **Files** tab is selected, displaying a file tree. The **root** directory is highlighted with a red box. A file explorer window is overlaid, showing a file named **input** being dragged from the local file system to the **root** directory of the container. A red arrow and the text **Drag and Drop** indicate the action.

Docker Environment for MIPS Compilation

- mips-compile /root/input.c

```
PS C:\Users\j\> docker run -it --rm boanlab/mips-ubuntu
root@7613a00ae2c3: /#
root@7613a00ae2c3: /# mips-compile /root/input.c
root@7613a00ae2c3: /#
```

- input.bin / input.s



> opt			13 days ago	drwxr-xr-x
> proc			1 minute ago	dr-xr-xr-x
✓ root	MODIFIED		13 days ago	drwx-----
.bashrc		3 kB	2 years ago	-rw-r--r--
.profile		161 Bytes	4 years ago	-rw-r--r--
input.bin	ADDED	112 Bytes	40 seconds ago	-rw-r--r--
input.c	ADDED	83 Bytes	24 minutes ago	-rwxr-xr-x
input.o	ADDED	1.2 kB	40 seconds ago	-rw-r--r--
input.s	ADDED	855 Bytes	40 seconds ago	-rw-r--r--
> run			13 days ago	drwxr-xr-x

Docker Environment for MIPS Compilation

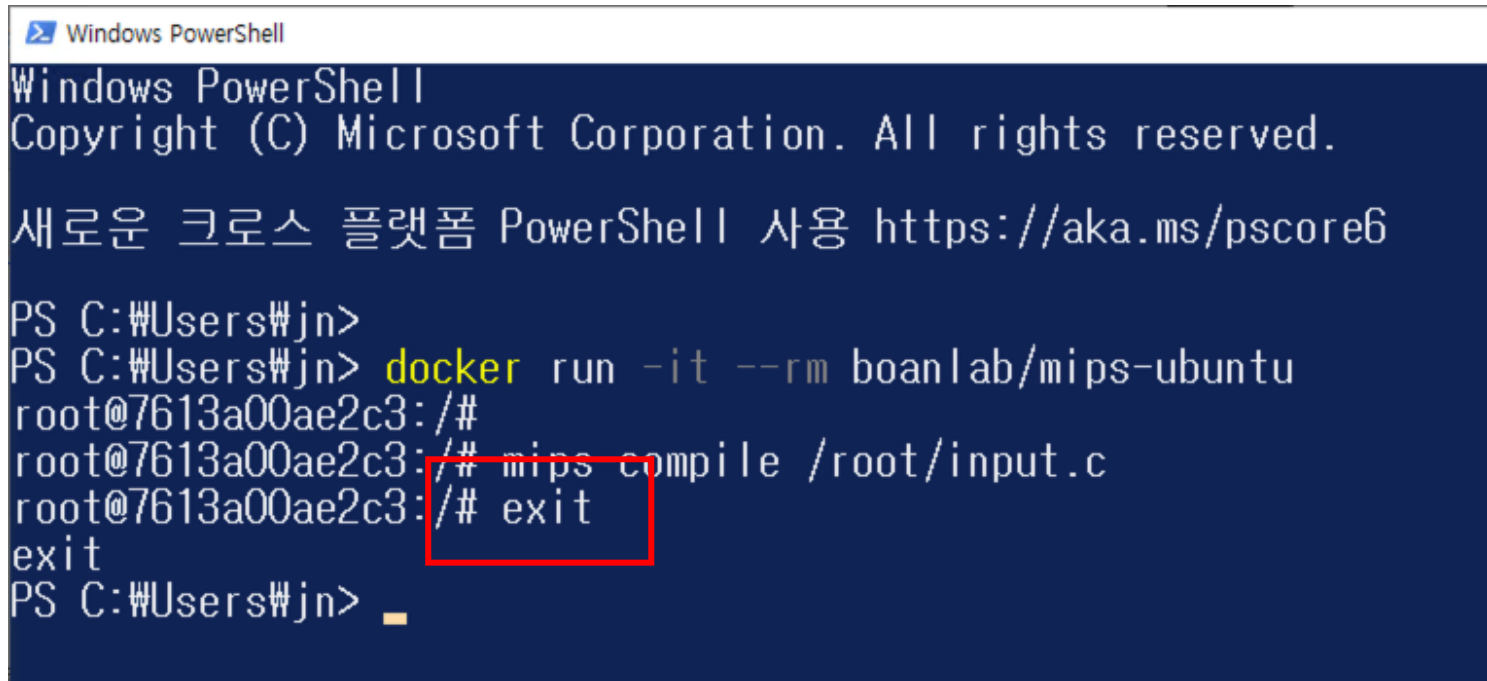
Docker Desktop interface showing a container named **brave_lehmann** (ID: 7613a00ae2c3) running the **boanlab/mips-ubuntu** image. The **Files** tab is selected, displaying a file list. A red box highlights the context menu for the file **input.bin**, showing options: **Edit file**, **Delete**, **Save**, and **Import**. A red text overlay reads "Save input.bin and input.s".

Name	Note	Size	Last modified	Mode
lib64 -> usr/lib64		9 Bytes	13 days ago	Lrwxrwxrwx
libx32 -> usr/libx32		10 Bytes	13 days ago	Lrwxrwxrwx
media			13 days ago	drwxr-xr-x
mnt			13 days ago	drwxr-xr-x
opt			13 days ago	drwxr-xr-x
proc			3 minutes ago	dr-xr-xr-x
root	MODIFIED		13 days ago	drwx-----
.bashrc		3 kB	2 years ago	-rw-r--r--
.profile		161 Bytes	4 years ago	-rw-r--r--
input.bin	ADDED	112 Bytes	2 minutes ago	-rw-r--r--
input.c	ADDED	83 Bytes	26 minutes ago	-rwxr-xr-x
input.o	ADDED	1.2 kB	2 minutes ago	-rw-r--r--
input.s	ADDED	855 Bytes	2 minutes ago	-rw-r--r--
run			13 days ago	drwxr-xr-x

RAM 6.83 GB CPU 3.15% Not connected to Hub v4.17.0

Docker Environment for MIPS Compilation

- exit



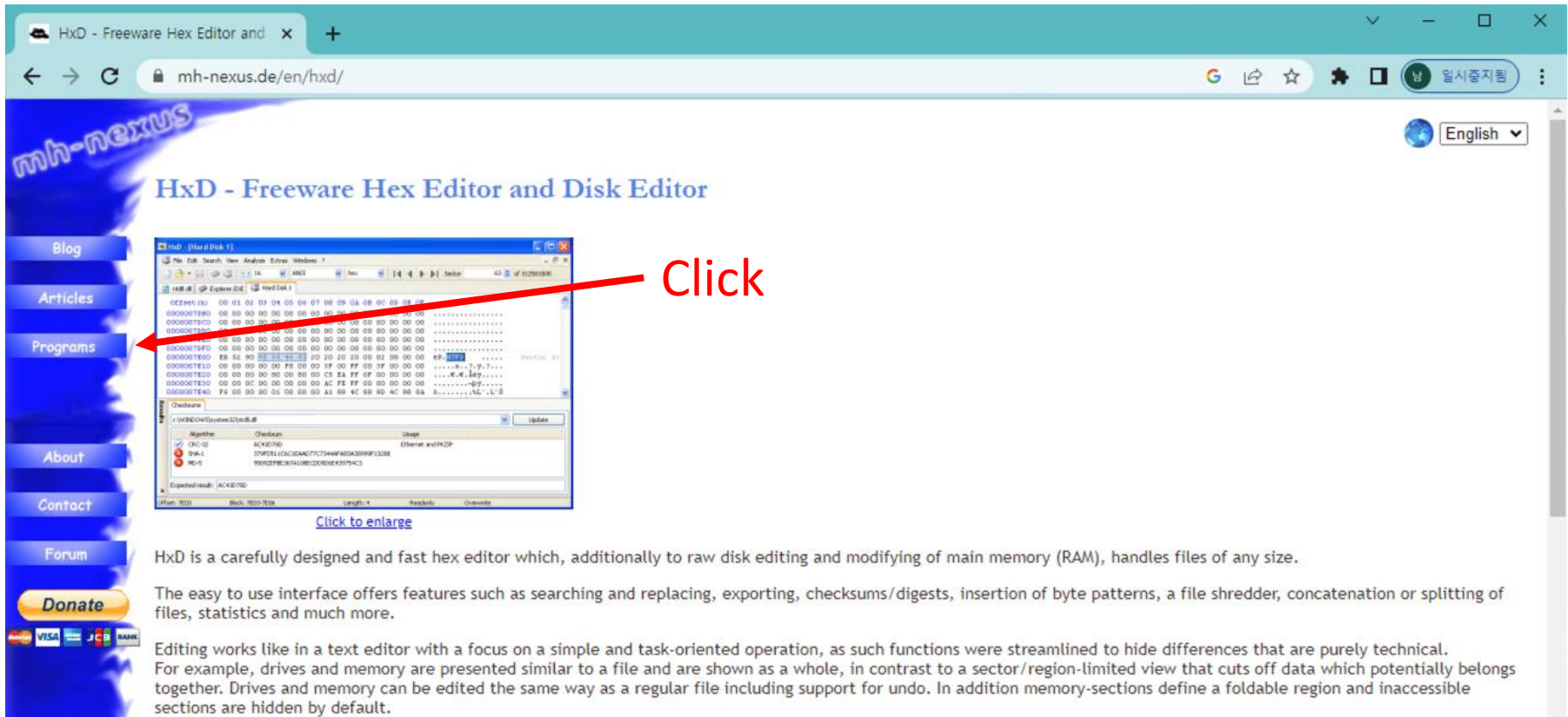
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\Wjn>
PS C:\Users\Wjn> docker run -it --rm boanlab/mips-ubuntu
root@7613a00ae2c3:/#
root@7613a00ae2c3:/# mips compile /root/input.c
root@7613a00ae2c3:/# exit
exit
PS C:\Users\Wjn> 
```

Fix Jump Address

- Install HxD



The screenshot shows a web browser window with the URL mh-nexus.de/en/hxd/. The page features a blue sidebar with navigation links: Blog, Articles, Programs, About, Contact, and Forum. A 'Donate' button is also present. The main content area is titled 'HxD - Freeware Hex Editor and Disk Editor'. It displays a preview of the HxD application interface, which includes a menu bar, a toolbar, and a hex editor window. A red arrow points to a button labeled 'Click' on the application icon. Below the preview, there is a 'Click to enlarge' link. The text below the image describes HxD as a fast hex editor that handles files of any size and offers features like searching, replacing, exporting, checksums/digests, and file shredding.

HxD is a carefully designed and fast hex editor which, additionally to raw disk editing and modifying of main memory (RAM), handles files of any size.

The easy to use interface offers features such as searching and replacing, exporting, checksums/digests, insertion of byte patterns, a file shredder, concatenation or splitting of files, statistics and much more.

Editing works like in a text editor with a focus on a simple and task-oriented operation, as such functions were streamlined to hide differences that are purely technical. For example, drives and memory are presented similar to a file and are shown as a whole, in contrast to a sector/region-limited view that cuts off data which potentially belongs together. Drives and memory can be edited the same way as a regular file including support for undo. In addition memory-sections define a foldable region and inaccessible sections are hidden by default.

Freeware Programs

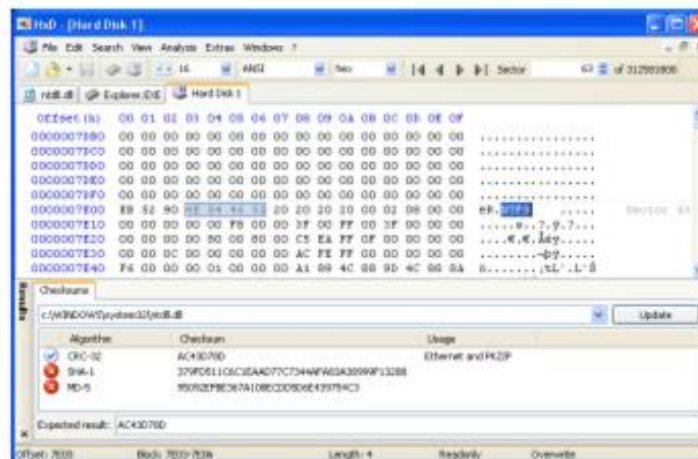
[Blog](#)
[Articles](#)
[Programs](#)
[About](#)
[Contact](#)
[Forum](#)
[Donate](#)


HxD

HxD is a carefully designed and fast hex editor which, additionally to raw disk editing and modifying of main memory (RAM), handles files of any size.


The easy to use interface offers features such as searching and replacing, exporting, checksums/digests, insertion of byte patterns, a file shredder, concatenation or splitting of files, statistics and much more.

[Read more...](#)



[Click to enlarge](#)

Click

Version	2.5.0.0 (February 11, 2021)	What's new?
OS	Windows XP, 2003, Vista, 7, 8 or 10	
	 Download page	

Fix Jump Address

- Download and extract a Zip file → Run the HxD installer

Click



 HxD20, Indonesian	installable	2.5.0.0	Syahriel Ibnu Irfansyah	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Italian	installable	2.5.0.0	Costantino Grana	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Japanese	installable	2.5.0.0	Airumu Zun	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Korean	installable	2.5.0.0	Jehwan Yun	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Dutch	installable	2.5.0.0	Jaap Kramer	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Polish	installable	2.5.0.0	Paweł Porwisz	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512
 HxD20, Portuguese (Brazil)	installable	2.5.0.0	Daniel Maganha	February 11, 2021	 Download per HTTPS 3.19 MiB	SHA-1 and SHA-512

Fix Jump Address

- Run HxD and open `input.bin`

16 → 4

HxD - [C:\Users\#jn\Downloads\input2\input.bin]

파일(F) 편집(E) 찾기(S) 보기(V) 분석(A) 도구(T) 형식(W) 도움말(H)

4 Windows (ANSI) 16진수

input.bin

Offset (h)	00	01	02	03	Decoded text
00000000	27	BD	FF	D8	'½ÿØ
00000004	AF	BF	00	24	—¿.\$
00000008	AF	BE	00	20	—¾.
0000000C	03	A0	F0	25	. ð%
00000010	24	02	00	04	\$. . .
00000014	AF	C2	00	1C	—Â..
00000018	8F	C4	00	1C	.Ä..
0000001C	0C	00	00	00
00000020	00	00	00	00
00000024	03	C0	E8	25	.Àè%
00000028	8F	BF	00	24	.¿.\$
0000002C	8F	BE	00	20	.¾.
00000030	27	BD	00	28	'½. (

특수 편집기

데이터 변환기

2진수 (8비트)	10101111
Int8	이동: -81
UInt8	이동: 175
Int16	이동: -16465
UInt16	이동: 49071
Int24	이동: 49071
UInt24	이동: 49071
Int32	이동: 604028847
UInt32	이동: 604028847
Int64	이동: 2306052668646277039
UInt64	이동: 2306052668646277039
LEB128	이동: 8111
ULEB128	이동: 8111
AnsiChar / char8_t	-
WideChar / char16_t	뽕
UTF-8 code point	유효하지 않은 Continuation Byte
Single (float32)	2.79179379349833E-17
Double (float64)	1.5611108860525E-154
OLETIME	1899-12-30
FILETIME	8908-08-04 오전 10:34:24

Fix Jump Address

- Open `input.s`

```
input - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

00000000 <main>:
0:    27bdffd8 addiu    sp,sp,-40
4:    afbf0024 sw      ra,36(sp)
8:    afbe0020 sw      s8,32(sp)
c:    03a0f025 move    s8,sp
10:   24020004 li      v0,4
14:   afc2001c sw      v0,28(s8)
18:   8fc4001c lw      a0,28(s8)
1c:   0c000000 jal      0 <main>
20:   00000000 nop
24:   03c0e825 move    sp,s8
28:   afbf0024 lw      ra,36(sp)
3c:   27bdffe0 addiu    sp,sp,-32
40:   afbf001c sw      ra,28(sp)
```

```
input - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

int foo(int index);

int main() {
    int index = 4;
    return foo(index);
}

int foo(int index) {
    if (index == 1)
        return 1;
    else
        return index + foo(index-1);
}
```

Should point 0x3c instead of 0x0

(Review) Implementing Jumps



- Jump uses word address
- Update PC with concatenation of
 - Top 4 bits of old PC
 - 26-bit jump address
 - 00



- Need an extra control signal decoded from opcode

Fix Jump Address

- Address for Jump = $(0x3c \gg 2) = 0xF$

Offset (h) 00 01 02 03 Decoded text

Offset (h)	00	01	02	03	Decoded text
00000000	27	BD	FF	D8	'½ÿØ
00000004	AF	BF	00	24	—¿. \$
00000008	AF	BE	00	20	—¾.
0000000C	03	A0	F0	25	. ð%
00000010	24	02	00	04	\$...
00000014	AF	C2	00	1C	—Â..
00000018	8F	C4	00	1C	.Ä..
0000001C	0C	00	00	0F
00000020	00	00	00	00
00000024	03	C0	E8	25	.Àè%
00000028	8F	BF	00	24	.¿. \$
0000002C	8F	BE	00	20	.¾.
00000030	27	BD	00	28	'½. (

00 → 0F

특수 편집기

데이터 변환기	
2진수 (8비트)	00001111
Int8	이동: 15
UInt8	이동: 15
Int16	이동: 15
UInt16	이동: 15
Int24	이동: 15
UInt24	이동: 15
Int32	이동: 15
UInt32	이동: 15
Int64	이동: -1675335762846941169
UInt64	이동: 16771408310862610447
LEB128	이동: 15
ULEB128	이동: 15
AnsiChar / char8_t	₩
WideChar / char16_t	₩
UTF-8 code point	₩ (U+000F)
Single (float32)	2.10194769648723E-44
Double (float64)	-3.74028881832669E196
OLETIME	유효하지 않음
FILETIME	유효하지 않음