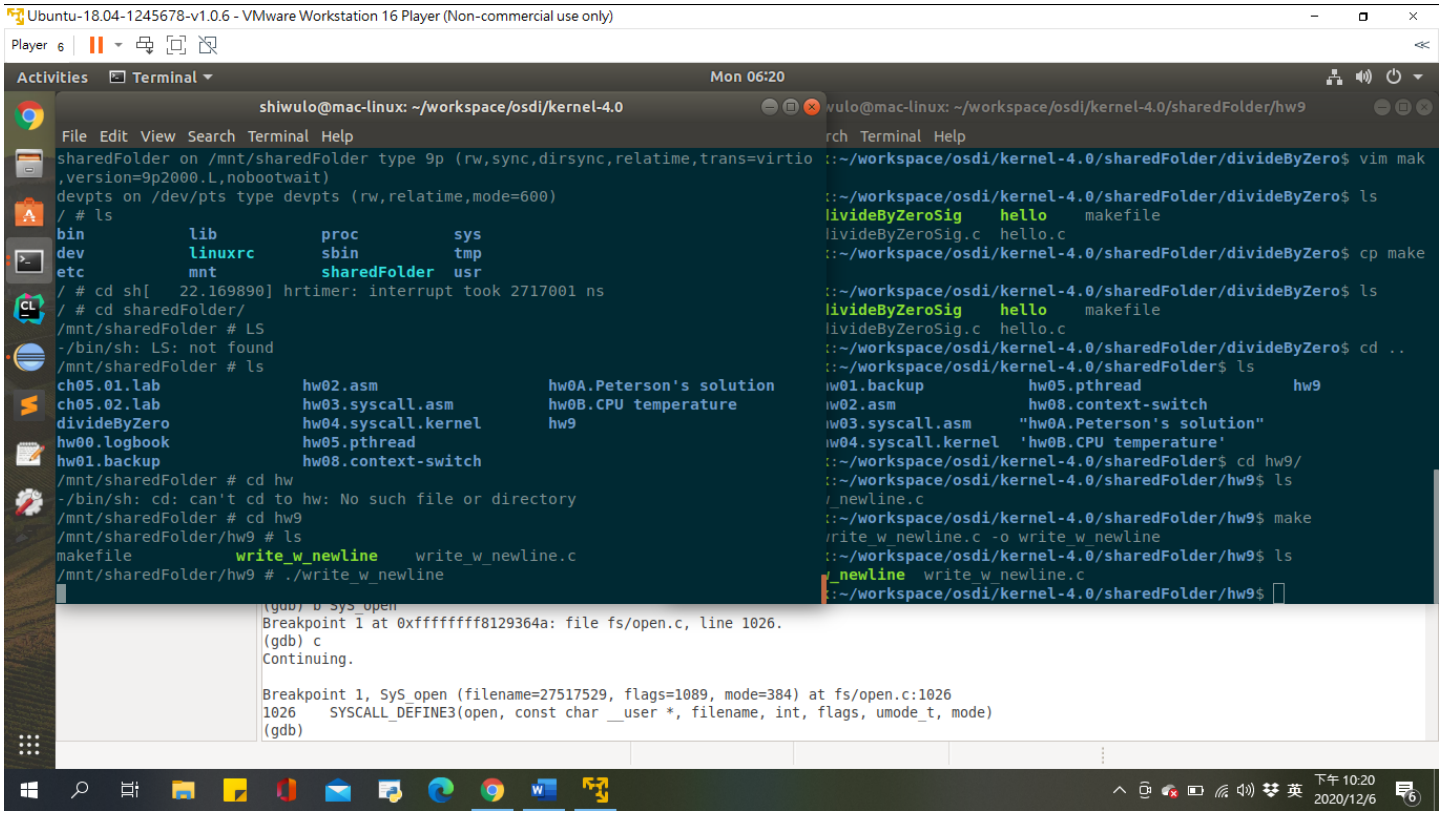
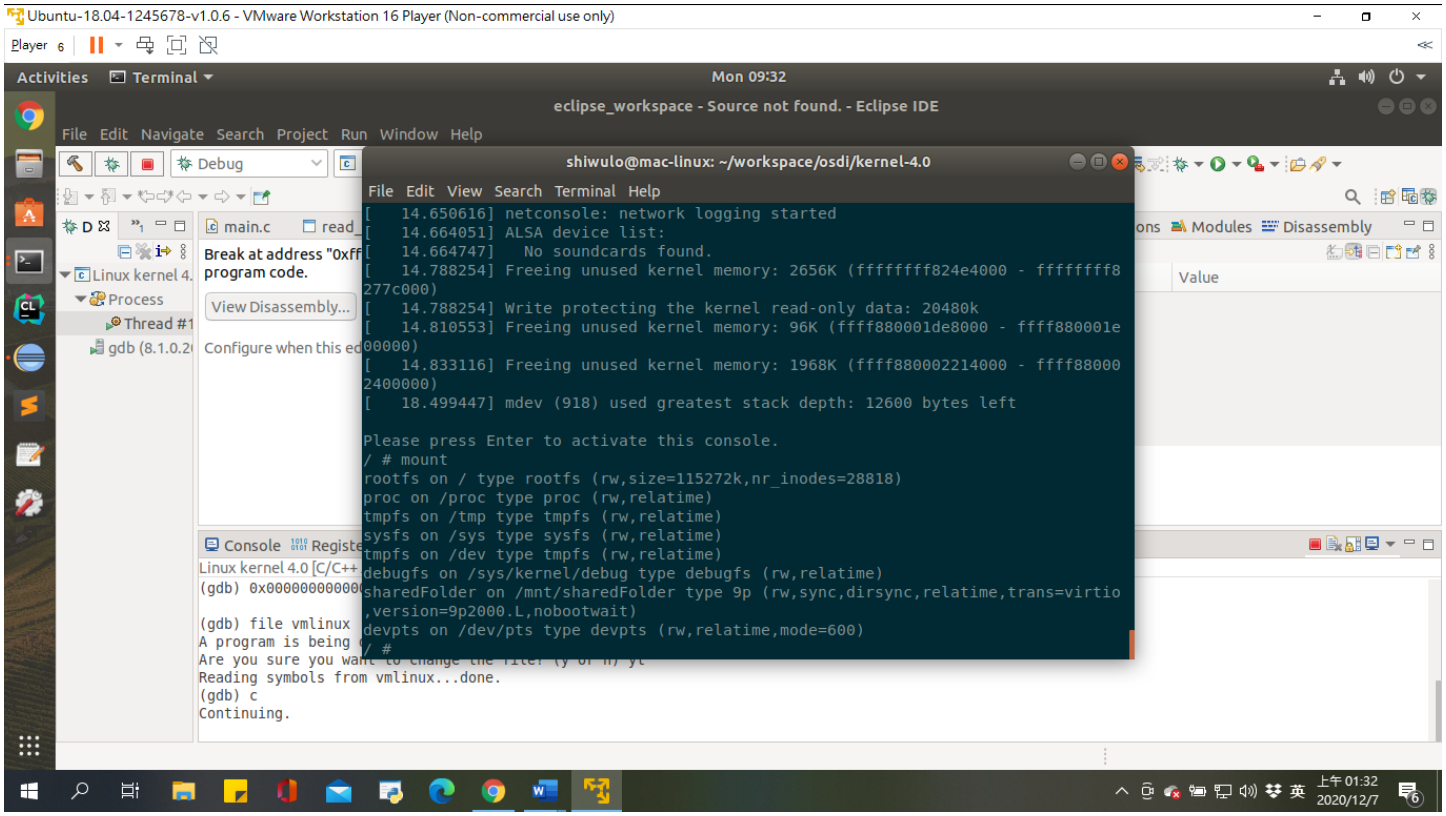
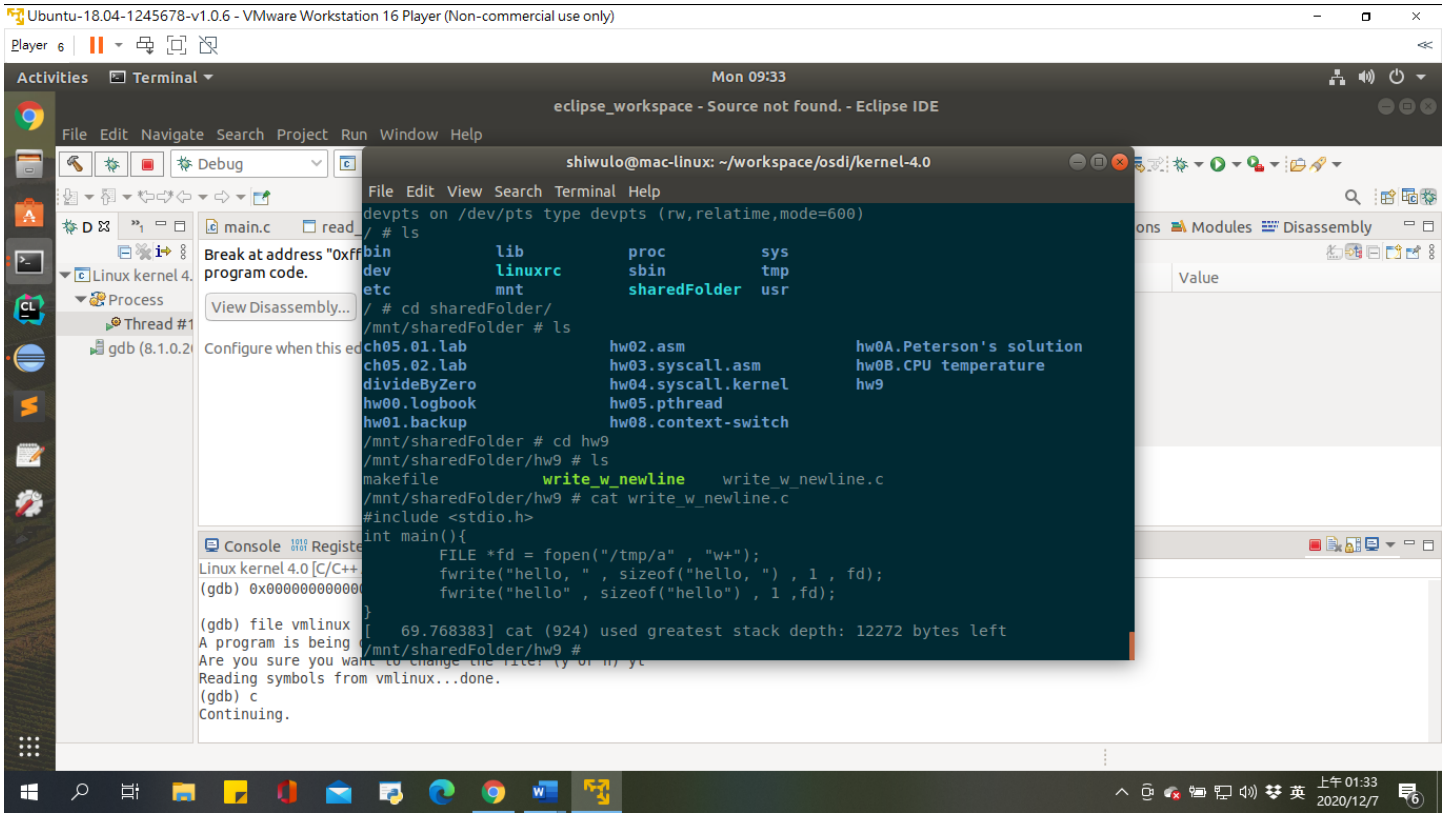
甲、

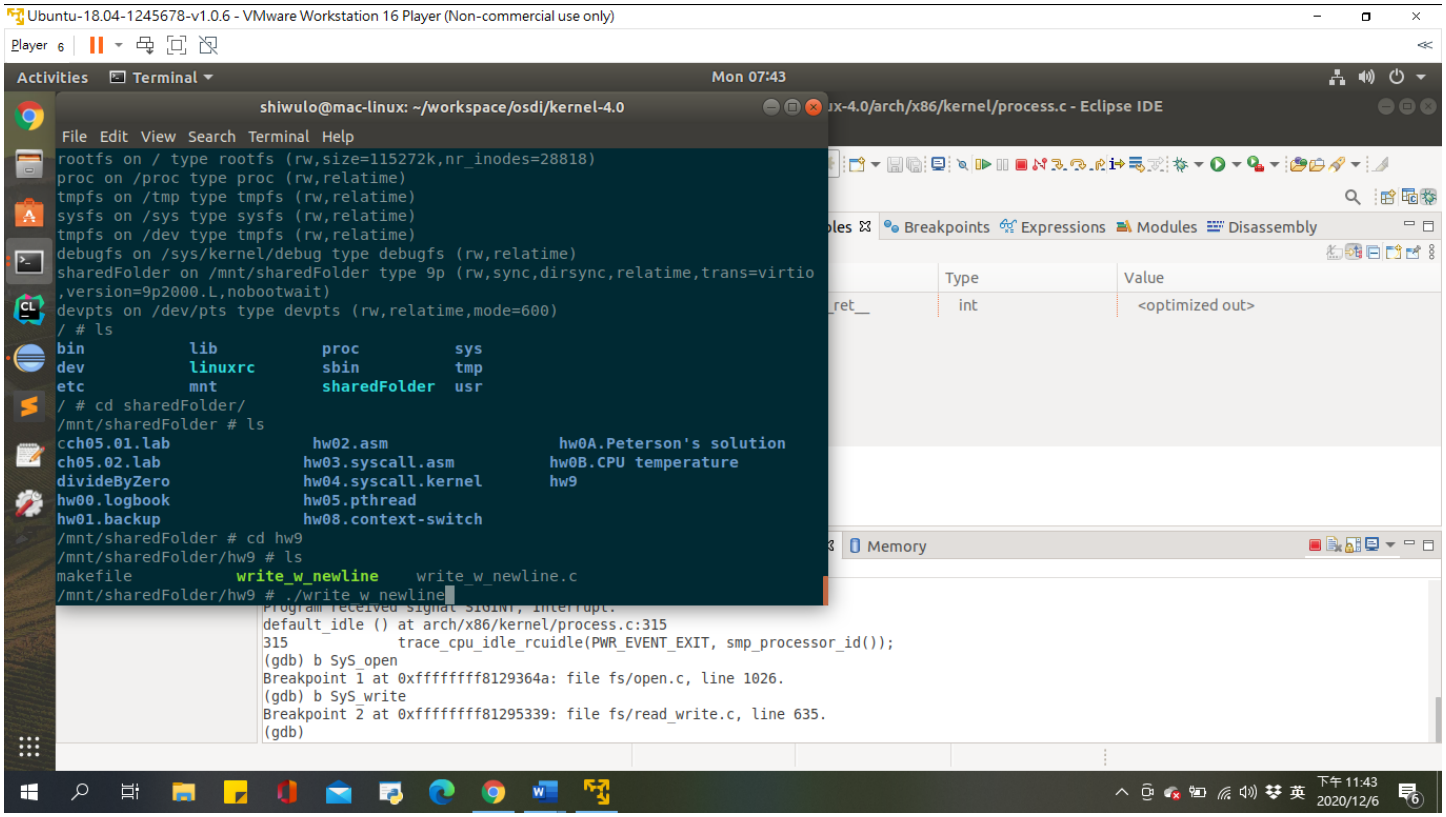
一、跟之前的作業方式一樣，使用 eclipse 連上我們的 GDB。




二、先將中斷點設在 b SyS_open、b SyS_write，執行預先寫好編譯好的 write_w_newline。




乙、

1. fdget_pos:將整數轉換成一個存取成一個對應的物件

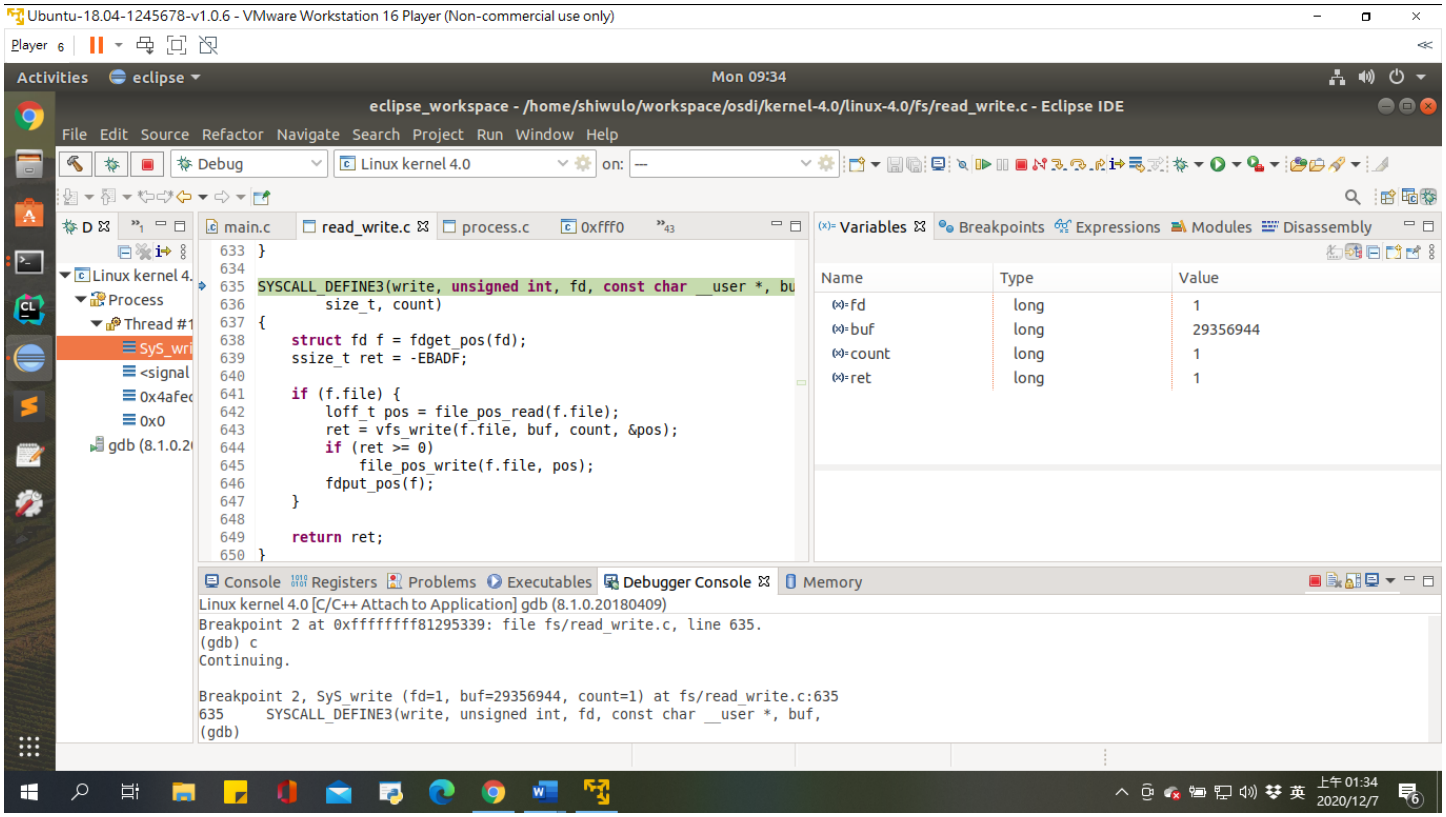
2. vfs_write:讓 FILE 找到相對應的 inode，並寫入定義的 write 中。

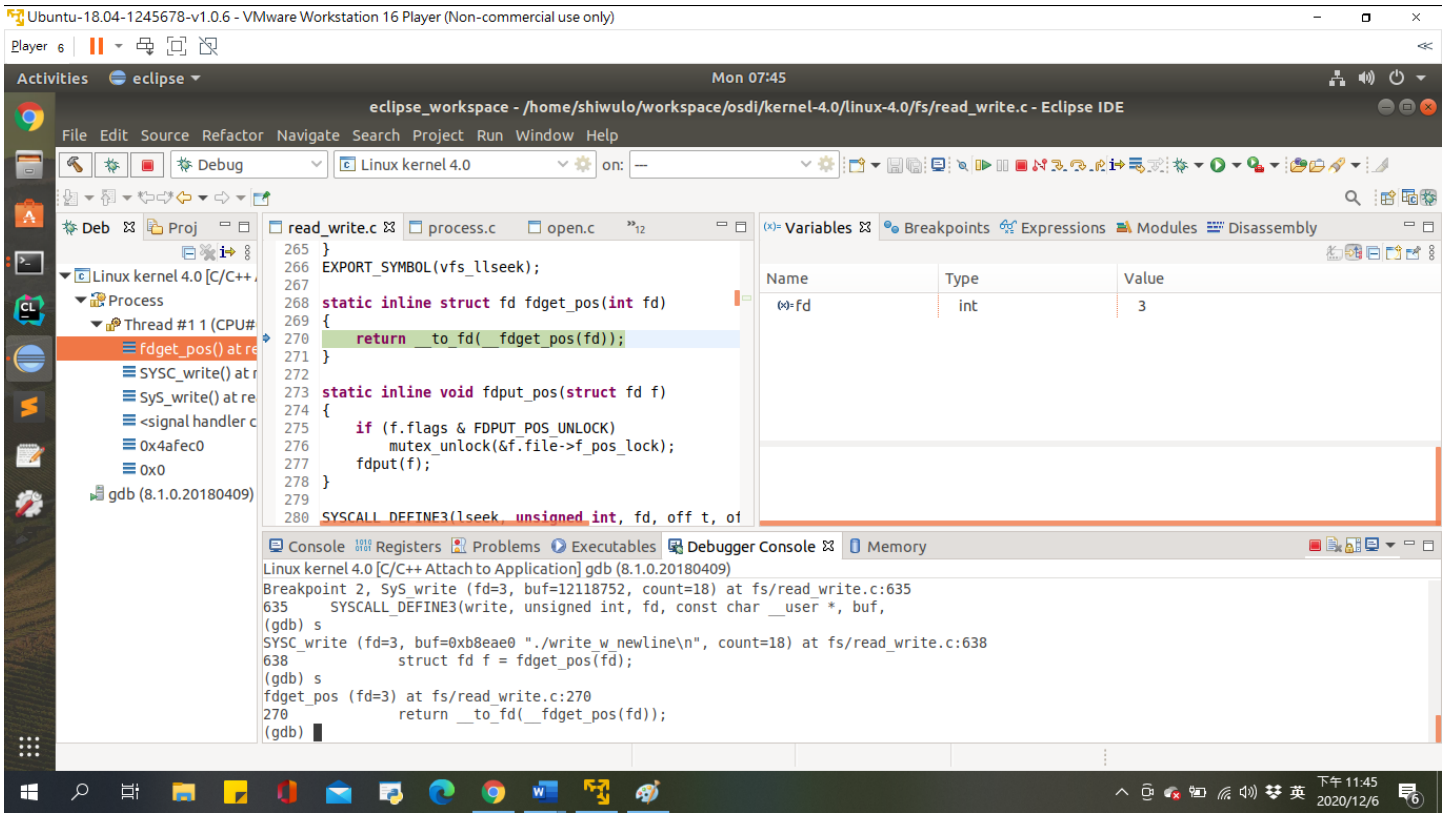
3. File_start_write:與 file_end_write 合起來就是__vfs_write

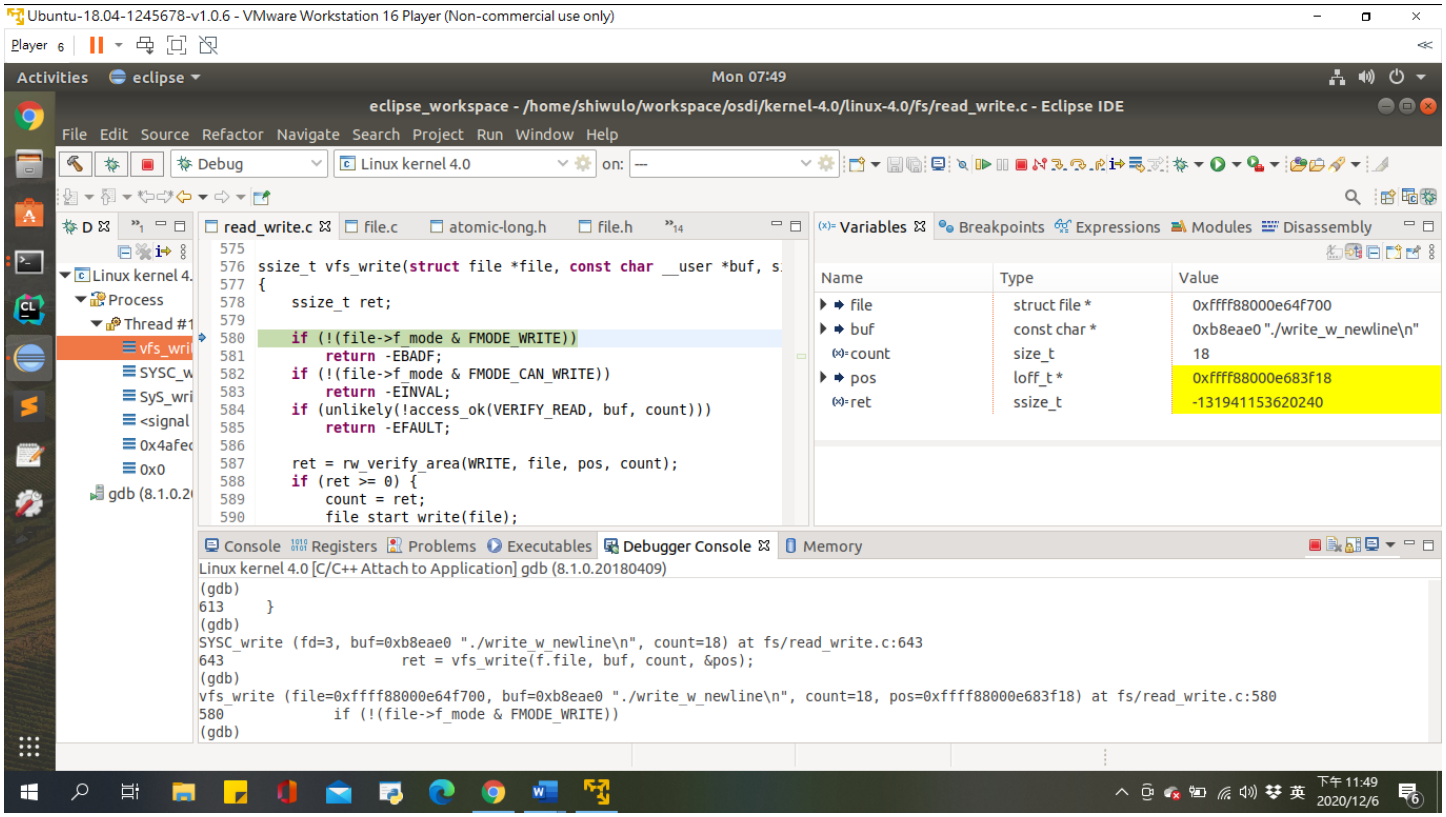
4. file_pos_write:將 pos 寫入 FILE 裡面

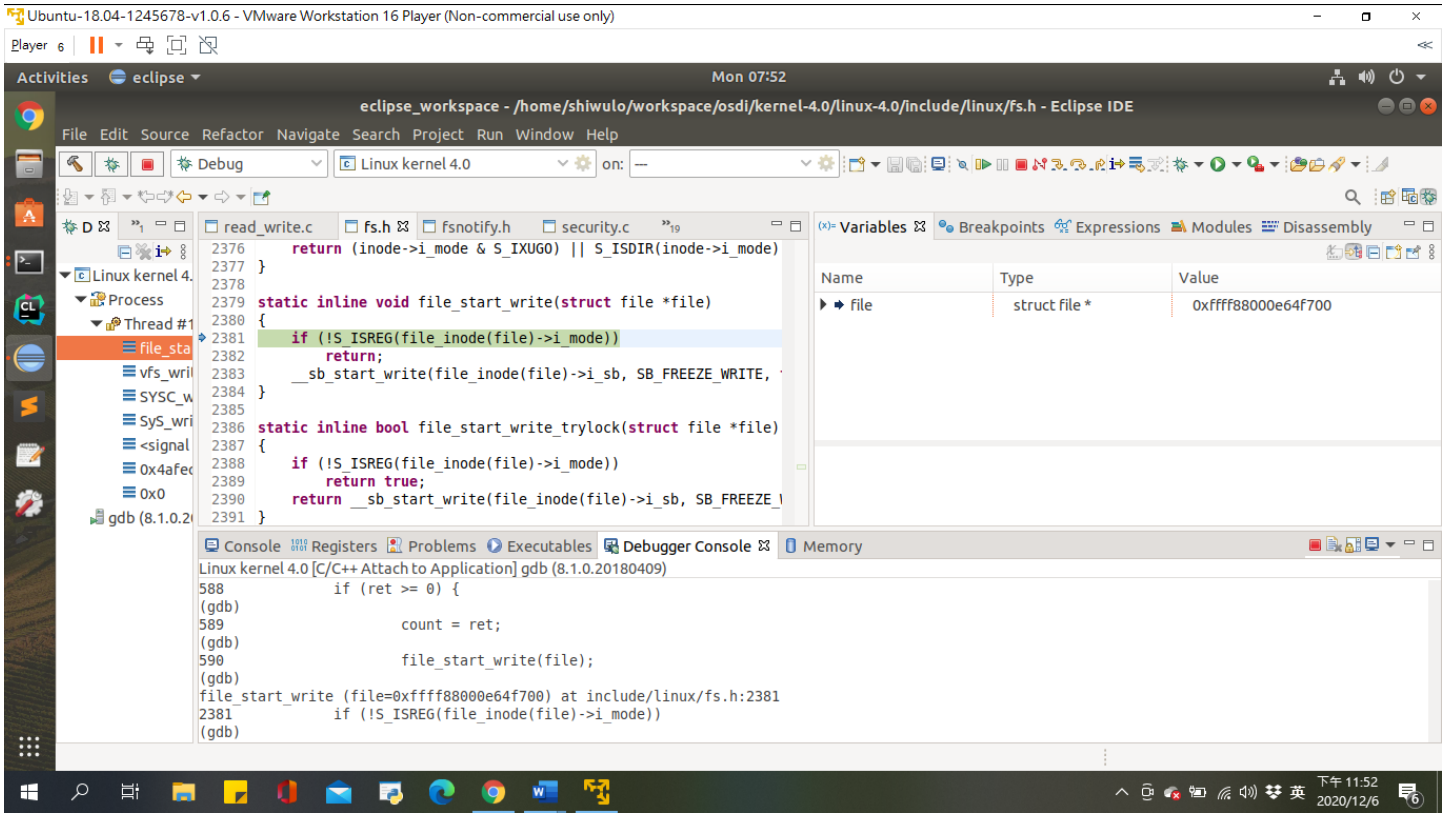
5. fdput_pos:印出結果

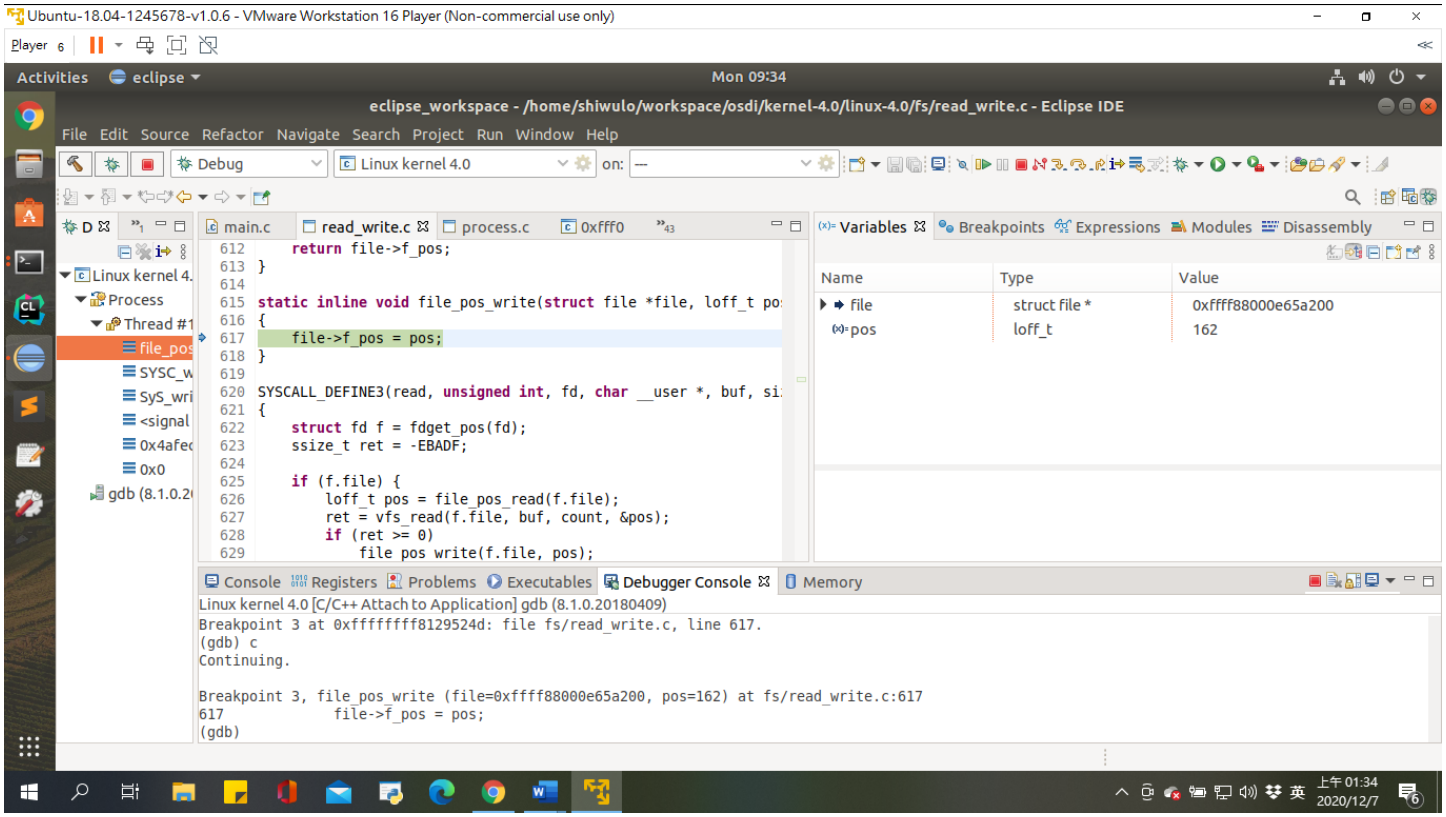
丙、

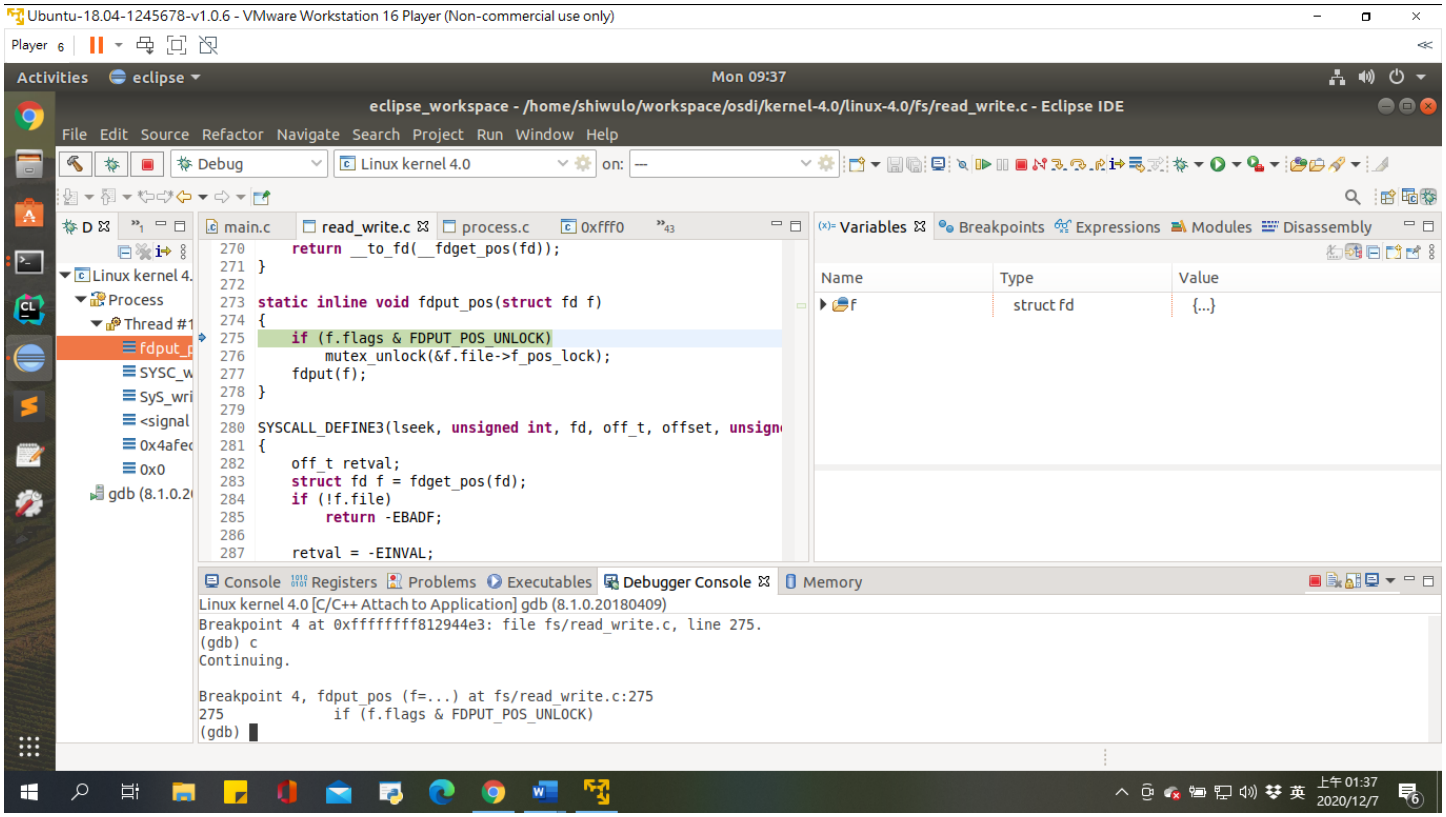


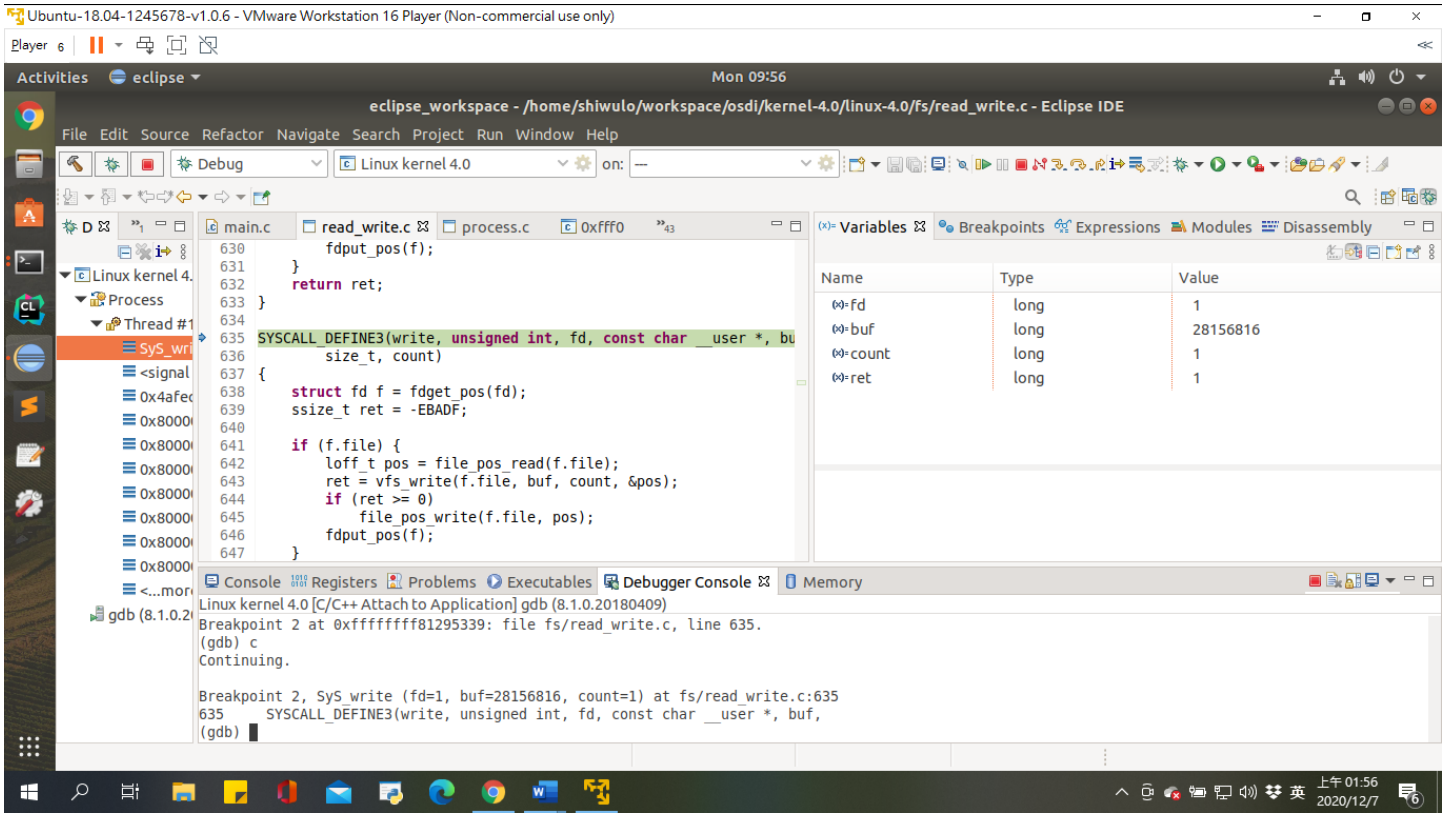

Ubuntu-18.04-1245678-v1.0.6 - VMware Workstation 16 Player (Non-commercial use only)
Activities
eclipse
Mon 07:49
eclipse_workspace - /home/shiwulo/workspace/osdi/kernel-4.0/linux-4.0/fs/read_write.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Linux kernel 4.0
read_write.c
file.c
atomic-long.h
file.h
575
576 ssize_t vfs_write(struct file *file, const char __user *buf, s
577 {
578 ssize_t ret;
579
580 if (!(file->f_mode & FMODE_WRITE))
581 return -EBADF;
582 if (!(file->f_mode & FMODE_CAN_WRITE))
583 return -EINVAL;
584 if (unlikely(!access_ok(VERIFY_READ, buf, count)))
585 return -EFAULT;
586
587 ret = rw_verify_area(WRITE, file, pos, count);
588 if (ret >= 0) {
589 count = ret;
590 file start write(file);
Variables
Breakpoints
Expressions
Modules
Disassembly
Name
Type
Value
file
struct file *
0xffff88000e64f700
buf
const char *
0xb8eae0 "./write_w_newline\n"
count
size_t
18
pos
loff_t *
0xffff88000e683f18
ret
ssize_t
-131941153620240
Console
Registers
Problems
Executables
Debugger Console
Memory
Linux kernel 4.0 [C/C++ Attach to Application] gdb (8.1.0.20180409)
(gdb)
613 }
(gdb)
SYSC_write (fd=3, buf=0xb8eae0 "./write_w_newline\n", count=18) at fs/read_write.c:643
643 ret = vfs_write(f.file, buf, count, &pos);
(gdb)
vfs_write (file=0xffff88000e64f700, buf=0xb8eae0 "./write_w_newline\n", count=18, pos=0xffff88000e683f18) at fs/read_write.c:580
580 if (!(file->f_mode & FMODE_WRITE))
(gdb)
Linux kernel 4.
Process
Thread #1
vfs_wri
SYSC_w
SyS_wri
<signal
0x4afec
0x0
gdb (8.1.0.2
下午 11:49
2020/12/6

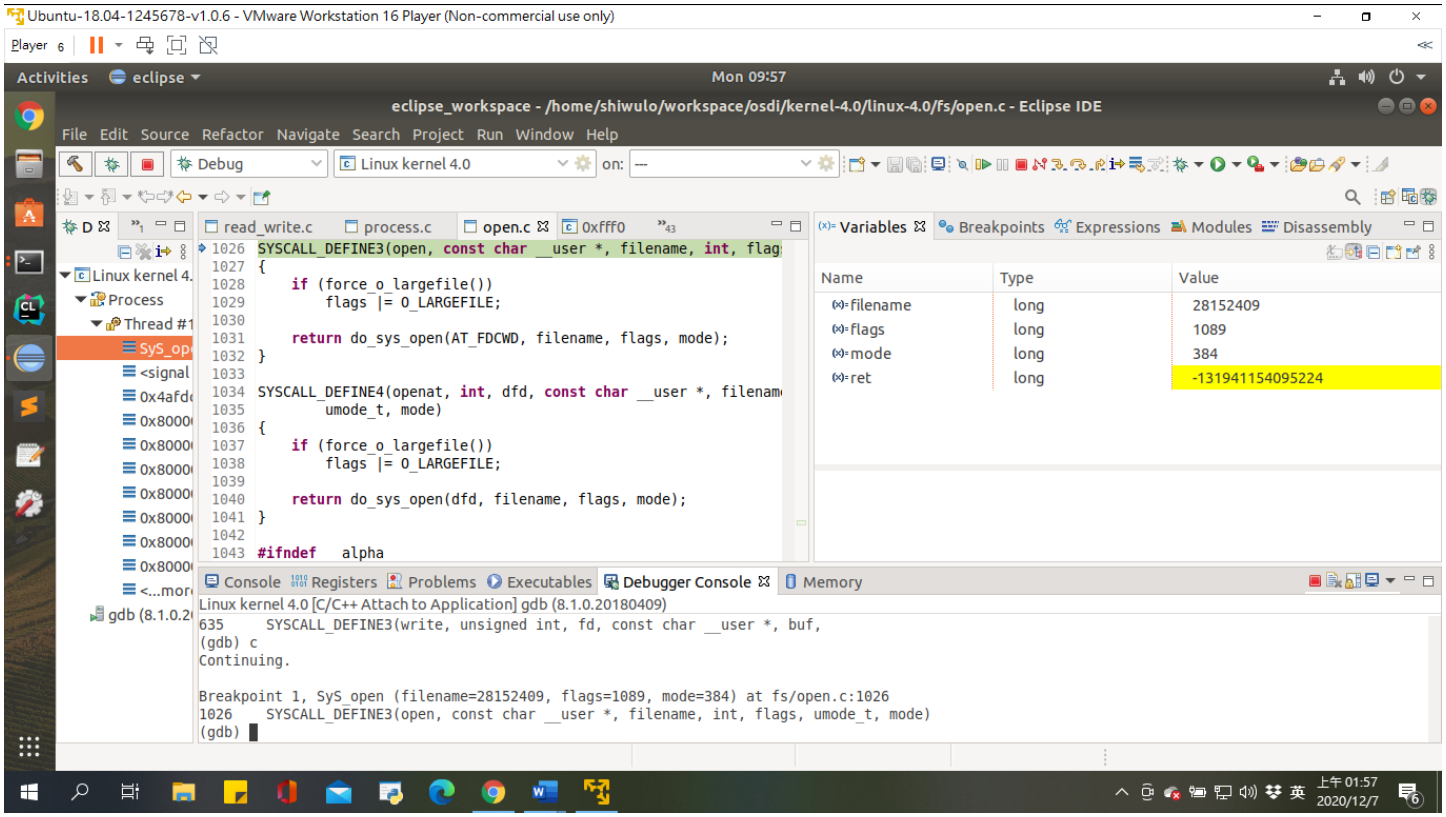

Ubuntu-18.04-1245678-v1.0.6 - VMware Workstation 16 Player (Non-commercial use only)
Activities
eclipse
Mon 07:52
eclipse_workspace - /home/shiwulo/workspace/osdi/kernel-4.0/linux-4.0/include/linux/fs.h - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Linux kernel 4.0
read_write.c
fs.h
fsnotify.h
security.c
2376 return (inode->i_mode & S_IXUGO) || S_ISDIR(inode->i_mode)
2377 }
2378
2379 static inline void file_start_write(struct file *file)
2380 {
2381 if (!S_ISREG(file_inode(file)->i_mode))
2382 return;
2383 __sb_start_write(file_inode(file)->i_sb, SB_FREEZE_WRITE,
2384 }
2385
2386 static inline bool file_start_write_trylock(struct file *file)
2387 {
2388 if (!S_ISREG(file_inode(file)->i_mode))
2389 return true;
2390 return __sb_start_write(file_inode(file)->i_sb, SB_FREEZE_
2391 }
Variables
Breakpoints
Expressions
Modules
Disassembly
Name
Type
Value
file
struct file *
0xffff88000e64f700
Console
Registers
Problems
Executables
Debugger Console
Memory
Linux kernel 4.0 [C/C++ Attach to Application] gdb (8.1.0.20180409)
588 if (ret >= 0) {
(gdb)
589 count = ret;
(gdb)
590 file_start_write(file);
(gdb)
file_start_write (file=0xffff88000e64f700) at include/linux/fs.h:2381
2381 if (!S_ISREG(file_inode(file)->i_mode))
(gdb)
Linux kernel 4.
Process
Thread #1
file_sta
vfs_wri
SYSC_w
SyS_wri
<signal
0x4afec
0x0
gdb (8.1.0.2
下午 11:52
2020/12/6

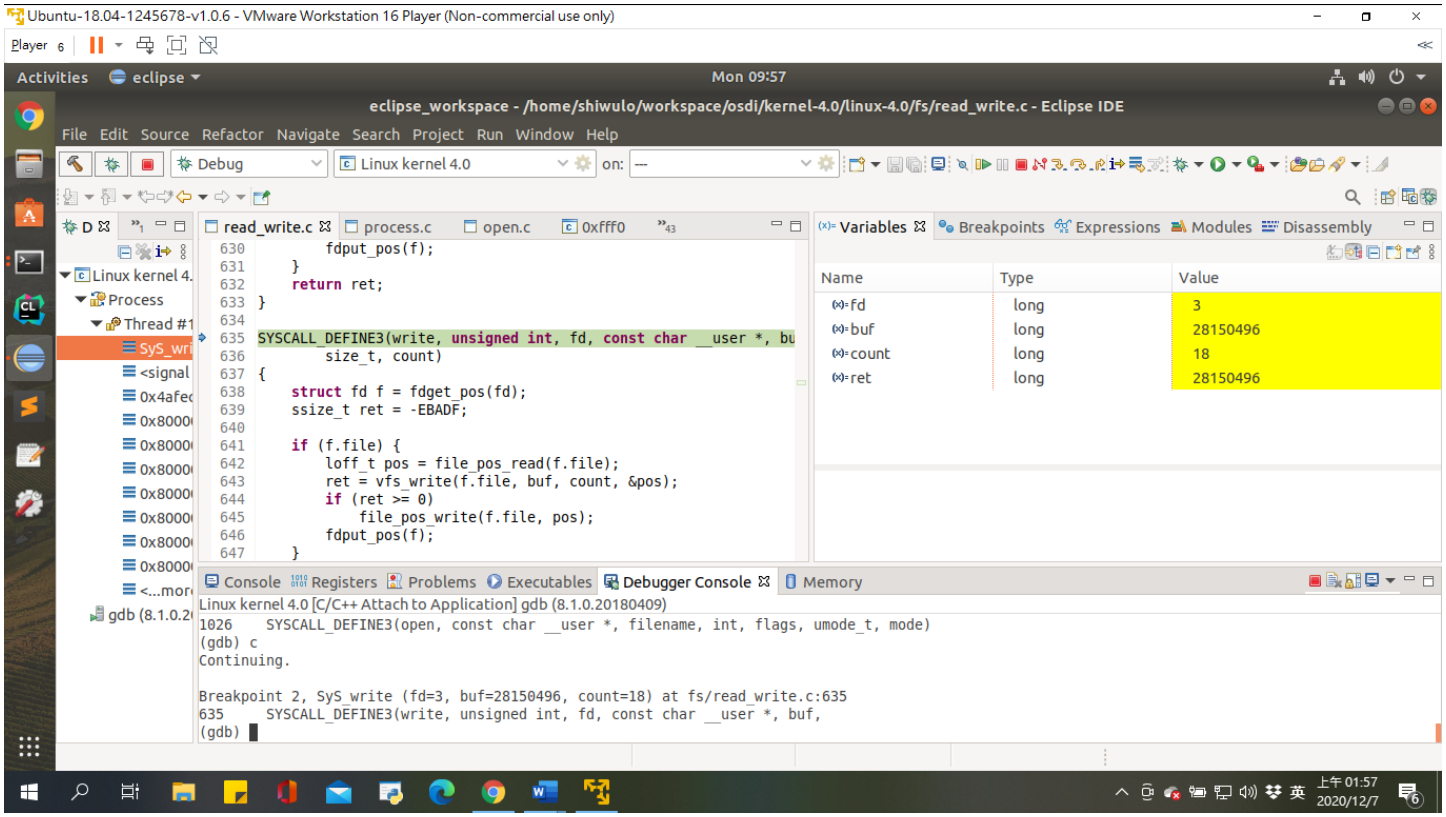



丁、應該在第二次進入SYSCALL_DEFINE3(write, **unsigned int**, fd, **const char** __user *, buf,size_t, count)，就進行寫入的動作。第一次write以及第一次open的時候，fd仍為1，直到第二次進入時，才發現他的值更動為3，由此可知當時才為真正SyS_write找到對應的實現點。


Ubuntu-18.04-1245678-v1.0.6 - VMware Workstation 16 Player (Non-commercial use only)
Activities
eclipse
Mon 09:56
eclipse_workspace - /home/shiwulo/workspace/osdi/kernel-4.0/linux-4.0/fs/read_write.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Linux kernel 4.0
main.c
read_write.c
process.c
0xfff0
Variables
Breakpoints
Expressions
Modules
Disassembly
Name
Type
Value
fd
long
1
buf
long
28156816
count
long
1
ret
long
1
630 fdput_pos(f);
631 }
632 return ret;
633 }
634
635 SYSCALL_DEFINE3(write, unsigned int, fd, const char __user *, bu
636 size_t, count)
637 {
638 struct fd f = fdget_pos(fd);
639 ssize_t ret = -EBADF;
640
641 if (f.file) {
642 loff_t pos = file_pos_read(f.file);
643 ret = vfs_write(f.file, buf, count, &pos);
644 if (ret >= 0)
645 file_pos_write(f.file, pos);
646 fdput_pos(f);
647 }
Console
Registers
Problems
Executables
Debugger Console
Memory
Linux kernel 4.0 [C/C++ Attach to Application] gdb (8.1.0.20180409)
Breakpoint 2 at 0xffffffff81295339: file fs/read_write.c, line 635.
(gdb) c
Continuing.
Breakpoint 2, SyS_write (fd=1, buf=28156816, count=1) at fs/read_write.c:635
635 SYSCALL_DEFINE3(write, unsigned int, fd, const char __user *, buf,
(gdb)
上午 01:56
2020/12/7


Ubuntu-18.04-1245678-v1.0.6 - VMware Workstation 16 Player (Non-commercial use only)
Activities
eclipse
Mon 09:57
eclipse_workspace - /home/shiwulo/workspace/osdi/kernel-4.0/linux-4.0/fs/open.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Linux kernel 4.0
read_write.c
process.c
open.c
0xfff0
Variables
Breakpoints
Expressions
Modules
Disassembly
Name
Type
Value
filename
long
28152409
flags
long
1089
mode
long
384
ret
long
-131941154095224
1026 SYSCALL_DEFINE3(open, const char __user *, filename, int, flag
1027 {
1028 if (force_o_largefile())
1029 flags |= O_LARGEFILE;
1030
1031 return do_sys_open(AT_FDCWD, filename, flags, mode);
1032 }
1033
1034 SYSCALL_DEFINE4(openat, int, dfd, const char __user *, filenam
1035 umode_t, mode)
1036 {
1037 if (force_o_largefile())
1038 flags |= O_LARGEFILE;
1039
1040 return do_sys_open(dfd, filename, flags, mode);
1041 }
1042
1043 #ifndef alpha
Console
Registers
Problems
Executables
Debugger Console
Memory
Linux kernel 4.0 [C/C++ Attach to Application] gdb (8.1.0.20180409)
635 SYSCALL_DEFINE3(write, unsigned int, fd, const char __user *, buf,
(gdb) c
Continuing.
Breakpoint 1, SyS_open (filename=28152409, flags=1089, mode=384) at fs/open.c:1026
1026 SYSCALL_DEFINE3(open, const char __user *, filename, int, flags, umode_t, mode)
(gdb)
上午 01:57
2020/12/7


Ubuntu-18.04-1245678-v1.0.6 - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
eclipse
Mon 09:57
eclipse_workspace - /home/shiwulo/workspace/osdi/kernel-4.0/linux-4.0/fs/read_write.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Linux kernel 4.0
on:
read_write.c
process.c
open.c
0xfff0
630 fdput_pos(f);
631 }
632 return ret;
633 }
634
635 SYSCALL_DEFINE3(write, unsigned int, fd, const char __user *, bu
636 size_t, count)
637 {
638 struct fd f = fdget_pos(fd);
639 ssize_t ret = -EBADF;
640
641 if (f.file) {
642 loff_t pos = file_pos_read(f.file);
643 ret = vfs_write(f.file, buf, count, &pos);
644 if (ret >= 0)
645 file_pos_write(f.file, pos);
646 fdput_pos(f);
647 }
Variables
Breakpoints
Expressions
Modules
Disassembly
Name
Type
Value
fd
long
3
buf
long
28150496
count
long
18
ret
long
28150496
Linux kernel 4.
Process
Thread #1
SyS_wri
<signal
gdb (8.1.0.2
Console
Registers
Problems
Executables
Debugger Console
Memory
Linux kernel 4.0 [C/C++ Attach to Application] gdb (8.1.0.20180409)
1026 SYSCALL_DEFINE3(open, const char __user *, filename, int, flags, umode_t, mode)
(gdb) c
Continuing.
Breakpoint 2, SyS_write (fd=3, buf=28150496, count=18) at fs/read_write.c:635
635 SYSCALL_DEFINE3(write, unsigned int, fd, const char __user *, buf,
(gdb)
上午 01:57
2020/12/7