VasabiEd โน๊ตเรื่อง การเรียนรู้ system programming อย่างง่ายๆ

การเขียนสร้างไฟล์และเขียนข้อมูล binary ลงสู่ไฟล์บนระบบ Linux วิธีการหนึ่งคือการใช้ open() system call เพื่อเปิดไฟล์ และใช้ write() system call เขียนข้อมูลลงในไฟล์นั้น

```
 1 #include <stdlib.h>
 2 #include <stdio.h>
 3 #include <string.h>
 4 #include <errno.h>
 5 #include <unistd.h>
 6 #include <stdint.h>
 7 #include <fcntl.h>
 8
 9 int qemu_write_full(int fd, const void *buf, size_t count);
10
11 #define         DATA_LENGTH     4096
12
13 main(int argc, char* argv[]){
14
15         int fd;
16         uint8_t *p;
17
18         fd = open("/tmp/mytest", O_WRONLY | O_CREAT, 0644);
19
20         p = malloc(sizeof(uint8_t) * DATA_LENGTH);
21
22         memset(p, 0, sizeof(uint8_t) * DATA_LENGTH);
23
24         qemu_write_full(fd, p, (sizeof(uint8_t) * DATA_LENGTH));
25
26         close(fd);
27
28 }
29
```

เมนโปรแกรม

บรรทัดที่ 18 เรียก open() system call เพื่อเปิดไฟล์แบบเพื่อเขียนข้อมูลลงอย่างเดียว ในกรณีที่ไม่มีไฟล์อยู่ก็ให้สร้างขึ้นด้วย (ใช้คำสั่ง man -s2 open เพื่อดูรายละเอียด) หลังจากนั้นโปรแกรม allocate เนื่อที่ใน memory ขนาด 4096 bytes และใส่ค่า 0 ลงไปโดยใช้ memset() system call

การเขียนข้อมูลลงไฟล์

เนื่องจากการเขียนข้อมูลนั้น write() system call อาจเขียนข้อมูลลงสู่ไฟล์ไม่หมดในการเรียกครั้งเดียวเพราะถูกขัดจังหวะ (interrupt) หรือ IO ไม่พร้อมใช้งานได้ ดังนั้นจึงต้องสร้าง wrapper funtion มาครอบเพื่อบังคับให้วนลูป

เขียนลงไฟล์ให้หมด (ฟังชั่นนี้ไม่เหมาะสมที่จะใช้เขียนข้อมูลลงไฟล์ที่ถูกเปิดแบบ O_NONBLOCK)

เนื่องจากผู้เขียนได้เอาตัวอย่าง wrapper function มาจาก source code ของ qemu open source software จึงได้คงชื่อ qemu_write_full() ไว้ในการเรียกใช้

```
30 // This function is taken from qemu's code. You may write
31 // your own using any name you like.
32
33 int qemu_write_full(int fd, const void *buf, size_t count){
34     ssize_t ret = 0;
35     ssize_t total = 0;
36
37     while (count) {
38         ret = write(fd, buf, count);
39         if (ret < 0) {
40             if (errno == EINTR)
41                 continue;
42             break;
43         }
44
45         count -= ret;
46         buf += ret;
47         total += ret;
48     }
49
50     return total;
51 }
```