

ICS-OS Lab 02

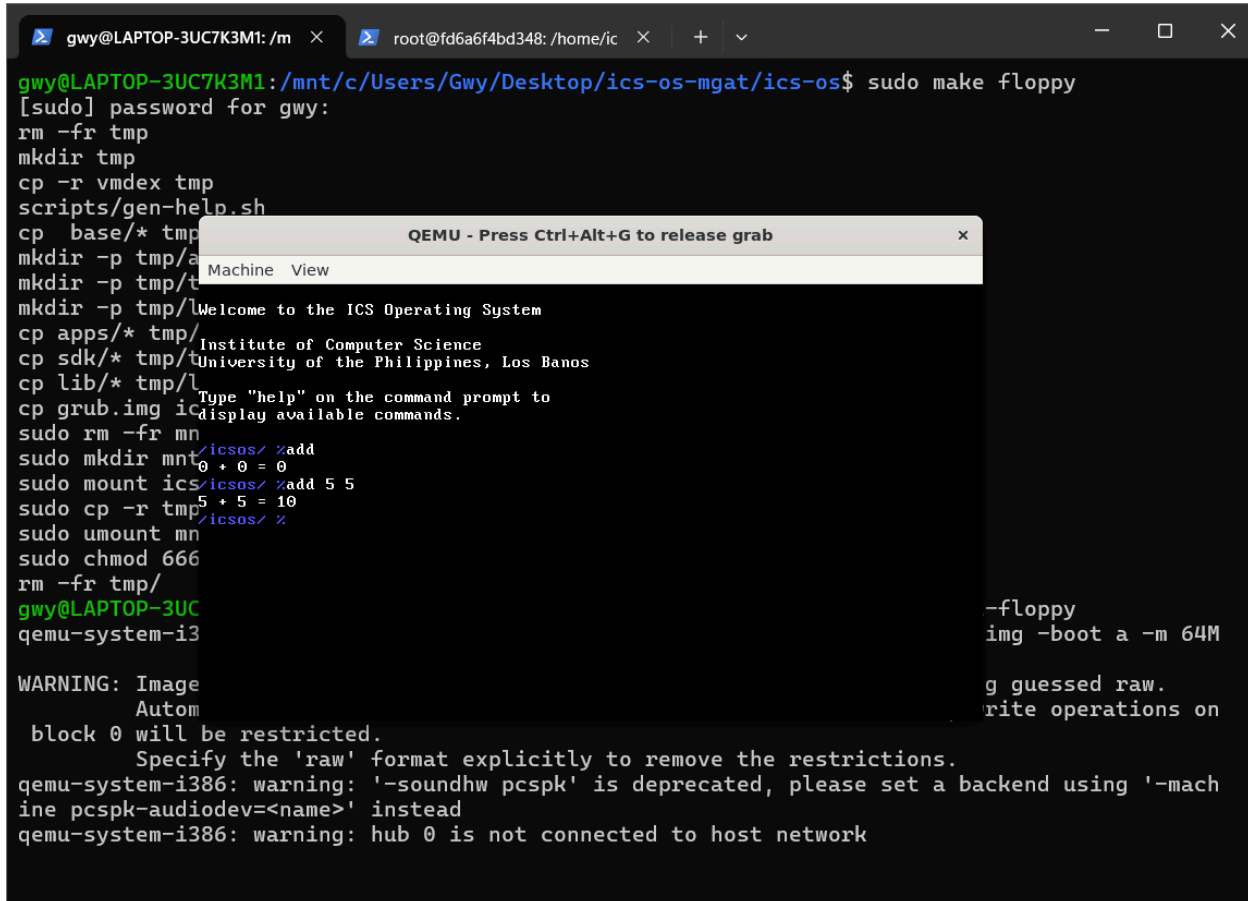
Command Line Interface, System Calls, and System Utilities

```
gwy@LAPTOP-3UC7K3M1: /m  ×  +  ▾
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat$ cd ics-os/
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ git checkout master
M       ics-os/base/icsos.hlp
M       ics-os/docker-compose.yml
M       ics-os/ics-os-floppy.img
Already on 'master'
Your branch is up to date with 'origin/master'.
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ git pull
Already up to date.
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ git checkout -b lab02
Switched to a new branch 'lab02'
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ git branch
  lab01
* lab02
  master
```

Task 1: Adding a new console command

```
C: > Users > Gwy > Desktop > ics-os-mgat > ics-os > kernel > console > C console.c
528  */
529  int console_execute(const char *str){
530      char temp[512];
531      char *u;
532      int command_length = 0;
533      signed char mouse_x, mouse_y, last_mouse_x=0, last_mouse_y=0;
534
535      //make a copy so that strtok wouldn't ruin str
536      strcpy(temp,str);
537      u=strtok(temp, " ");
538
539      if (u == 0)
540          return;
541
542      command_length = strlen(u);
543      if (strcmp(u,"add") == 0){ //-- Adds two integers. Args: <num1> <num2>
544          int a, b;
545          u = strtok(0, " ");
546          a = atoi(u);
547          u = strtok(0, " ");
548          b = atoi(u);
549          printf("%d + %d = %d\n",a,b,a+b);
550      }else
551          //check if a pathcut command was executed
552          if (u[command_length - 1] == ':'){
```

Build and boot ICS-OS to test if the command works



```
gwyy@LAPTOP-3UC7K3M1: /mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ sudo make floppy
[sudo] password for gwyy:
rm -fr tmp
mkdir tmp
cp -r vmdex tmp
scripts/gen-help.sh
cp base/* tmp
mkdir -p tmp/a
mkdir -p tmp/t
mkdir -p tmp/Welcome to the ICS Operating System
cp apps/* tmp/
cp sdk/* tmp/
cp lib/* tmp/
cp grub.img ic
sudo rm -fr mn
sudo mkdir mnt
sudo mount ics/mnt
sudo cp -r tmp/
sudo umount mn
sudo chmod 666
rm -fr tmp/
gwyy@LAPTOP-3UC
qemu-system-i386

WARNING: Image
Autom
block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.
qemu-system-i386: warning: '-soundhw pcspk' is deprecated, please set a backend using '-mach
ine pcspk-audiodev=<name>' instead
qemu-system-i386: warning: hub 0 is not connected to host network

QEMU - Press Ctrl+Alt+G to release grab
Machine View

-floppy
img -boot a -m 64M
g guessed raw.
rite operations on
```

QUESTION: What are the advantages and disadvantages of having the CLI as part of the kernel itself instead of a user application like Bash?

One advantage of having the CLI part of the kernel is that it speeds up communication with the kernel. This means that system calls would also be faster. Since CLI is part of the kernel, other kernel modules can add their own custom commands to it. The CLI is usually kept separate from the kernel because it can be changed, which is one of its disadvantages. Another reason is that if the CLI crashes, it will affect the whole kernel because they are linked.

Task 2: Adding a new system call service/function

Add the function

```
C: > Users > Gwy > Desktop > ics-os-mgat > ics-os > kernel > dexapi > C dex32API.c
23  *****/
24
25  #include "dex32API.h"
26
27  int dex32_getversion(){
28      return DEX32_OSVER;
29  };
30
31  int kchown(int fd, int uid, int gid){
32      printf("Changing owner of fd=%d to user id=%d and group id=%d\n", fd, uid, gid);
33      //Actual code to change file ownership is placed here.
34      //For now this is just empty
35      return 0; //0-success
36  }
```

Add it to the system call table

```
88      //api_addsystemcall(0xB,createthread,0,0);
89      api_addsystemcall(0xC,dex32_wait,0,0);
90      api_addsystemcall(0xC2, kchown, 0, 0);
91      api_addsystemcall(0xD,textcolor,0,0);
```

Task 3: Invoking a system call in a system utility

Task 3a: Creating the source

```
gwy@LAPTOP-3UC7K3M1: /m  X  root@fd6a6f4bd348: /home/ic  X  +  v
sudo cp -r tmp/* mnt/
sudo umount mnt
sudo chmod 666 ics-os-floppy.img
rm -fr tmp/
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ make boot-floppy
qemu-system-i386 -net nic,model=rtl8139 -soundhw pcspk -fda ics-os-floppy.img -boot a -m 64M

WARNING: Image format was not specified for 'ics-os-floppy.img' and probing guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
Specify the 'raw' format explicitly to remove the restrictions.
qemu-system-i386: warning: '-soundhw pcspk' is deprecated, please set a backend using '-machine pcspk-audiodev=<name>' instead
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ cd contrib
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os/contrib$ cp -r hello/ chown/
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os/contrib$ cd chown/
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os/contrib/chown$ mv hello.c chown.c
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os/contrib/chown$ sed -i 's/hello/chown/g' Makefile
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os/contrib/chown$ |
```

Task 3b: Building the executable and install

```
gwy@LAPTOP-3UC7K3M1: /mr X root@fd6a6f4bd348: /home/i X + v
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g -o
devmgr_error.o devmgr/devmgr_error.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g -o
kernel32.o kernel32.c
nasm -f elf32 -o startup.o startup/startup.asm
nasm -f elf32 -o asmlib.o startup/asmlib.asm
startup/asmlib.asm:321: warning: label alone on a line without a colon might be in error
nasm -f elf32 -o irqwrap.o irqwrap.asm
#strip --strip-debug *.o
ld -melf_i386 -T lscript.ld -Map mapfile.txt
objcopy --only-keep-debug Kernel32.bin Kernel32.sym
objcopy --strip-debug Kernel32.bin
gzip -c -9 Kernel32.bin > vmdex
cp vmdex ..
root@fd6a6f4bd348:/home/ics-os# cd contrib/chown
root@fd6a6f4bd348:/home/ics-os/contrib/chown# make
gcc -m32 -w -nostdlib -fno-builtin -static -ochown.exe chown.c ../../sdk/tccsdk.c ../../sdk/lib
tcc1.c ../../sdk/crt1.c
root@fd6a6f4bd348:/home/ics-os/contrib/chown# make install
cp chown.exe ../../apps
root@fd6a6f4bd348:/home/ics-os/contrib/chown#
```

Task 3c: Running the executable inside ICS-OS

```
48:/home/ics-os/contrib/chown# exit
K3
f Machine View
Institute of Computer Science
University of the Philippines, Los Banos

Type "help" on the command prompt to
display available commands.

/icsos/ %cd apps
ps/icsos/apps/ %ls -l -oname
Filename Size(bytes) Attribute Date Modified
c1. 0 d-rw 11/26/2024
b1. 0 d-rw 11/26/2024
pp chown.exe 20216 -xrw 11/26/2024
c1 ed.exe 53194 -xrw 11/26/2024
b1 hello.exe 20120 -xrw 11/26/2024
lz dmp.exe 20236 -xrw 11/26/2024
-n lzozip.exe 24024 -xrw 11/26/2024
os nasm.exe 308736 -xrw 11/26/2024
* pak.exe 17099 -xrw 11/26/2024
tcc.exe 186056 -xrw 11/26/2024
vgademo.exe 20472 -xrw 11/26/2024
Total Files: 11 Total Size: 670153 bytes
/icsos/apps/ %chown.exe
Hello World from ICS-OS!
/icsos/apps/ %
```

QUESTION: What is the output after executing chown.exe inside ICS-OS? The same as hello.exe

Task 3d: Modifying chown.c to invoke the new service via syscall

```
C: > Users > Gwy > Desktop > ics-os-mgat > ics-os > contrib > chown > C chown.c
1  #include "../sdk/dexsdk.h"
2  #define KCHOWN_SERVICE_NO 0xC2
3  int main(int argc, char *argv[]) {
4      if (argc < 4){
5          printf("Usage: chown.exe <fd> <uid> <gid>\n");
6          return -1;
7      }
8      dexsdk_systemcall(KCHOWN_SERVICE_NO, atoi(argv[1]), atoi(argv[2]),
9          atoi(argv[3]), 0, 0);
10     return 0;
11 }
```



```
C7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ sudo make floppy
QEMU
Machine View
/icsos/apps/ %ls -l -oname
Filename                               Size(bytes)  Attribute  Date Modified
.                                       0            d-rw      11/26/2024
..                                      0            d-rw      11/26/2024
chown.exe                             20216        -xrw      11/26/2024
ed.exe                                53194        -xrw      11/26/2024
hello.exe                             20120        -xrw      11/26/2024
hxdmp.exe                             20236        -xrw      11/26/2024
lzzip.exe                             24024        -xrw      11/26/2024
nasm.exe                              308736       -xrw      11/26/2024
pak.exe                               17099        -xrw      11/26/2024
tcc.exe                               186056       -xrw      11/26/2024
vgademo.exe                           20472        -xrw      11/26/2024
Total Files: 11  Total Size: 670153 bytes
/icsos/apps/ %chown.exe
Usage: chown.exe <fd> <uid> <gid>
/icsos/apps/ %_
C7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$
386 -net nic,model=rtl8139 -soundhw pcspk -fda ics-os-floppy.img
```

QUESTION: Study the function `dexsdk_systemcall()` defined in `sdktccsdk.c`. What does this function do? Discuss two other functions that call `dexsdk_systemcall()`.

From what I've seen, `dexsdk_systemcall()` is the most-used function in the program. It handles the basic control functions of the CLI, like moving the cursor, setting the screen's x and y coordinates, and clearing the screen. Two functions that use `dexsdk_systemcall()` are (a) `get_date_time()`, which tells the user what time it is and what day it is, and (b) `clrscr()`, which just clears the terminal screen.

QUESTION: What is the output of executing `chown.exe`, complete with command line arguments, this time? Describe what you think happened.

```
/icsos/apps/ %chown.exe
Usage: chown.exe <fd> <uid> <gid>
/icsos/apps/ %_
```

The program `chown.exe` did not perform the `dexsdk_systemcall` because the required arguments (`fd`, `uid`, `gid`) were not provided. Instead, it displayed the usage instructions.

Task 4: Cleanup

```
cp chown.exe ../../apps
root@be1286d9f915:/home/ics-os/contrib/chown# exit
exit
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ git checkout master
M       ics-os/base/icsos.hlp
M       ics-os/docker-compose.yml
M       ics-os/ics-os-floppy.img
M       ics-os/kernel/console/console.c
M       ics-os/kernel/dexapi/dex32API.c
M       ics-os/kernel/mapfile.txt
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ |
```

REFLECTION

This lab allowed me to understand how operating systems work, especially how command-line interfaces (CLIs) and system calls interact with the kernel. Adding a new system call to the kernel gave me insight into how the kernel exposes its capabilities to user applications. It made me realize how important it is to organize and handle system functions carefully. The process of defining the syscall, adding it to the syscall table, and then invoking it in a utility like `chown.exe` helped me see the structured communication between user space and kernel space. Reusing low-level functions like `dexsdk_systemcall()` to perform various tasks was intriguing, as it highlighted the importance of efficient and reusable code. One question that came to mind was, “How do modern systems make their CLI faster and safer at the same time?” It was satisfying to see how small changes to the kernel, system calls, and utilities manifest at the user interface level. This made me appreciate the design and organization required in operating systems. Overall, working in this lab taught me how hard it is to make operating systems that are fast, secure, and perform well.