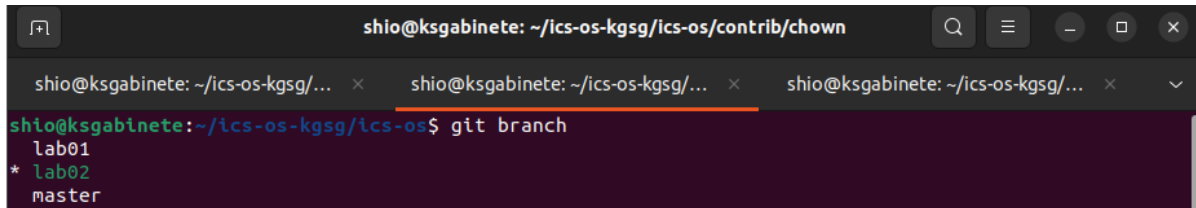


ICS-OS Lab 02: Command Line Interface, System Calls, and System Utilities

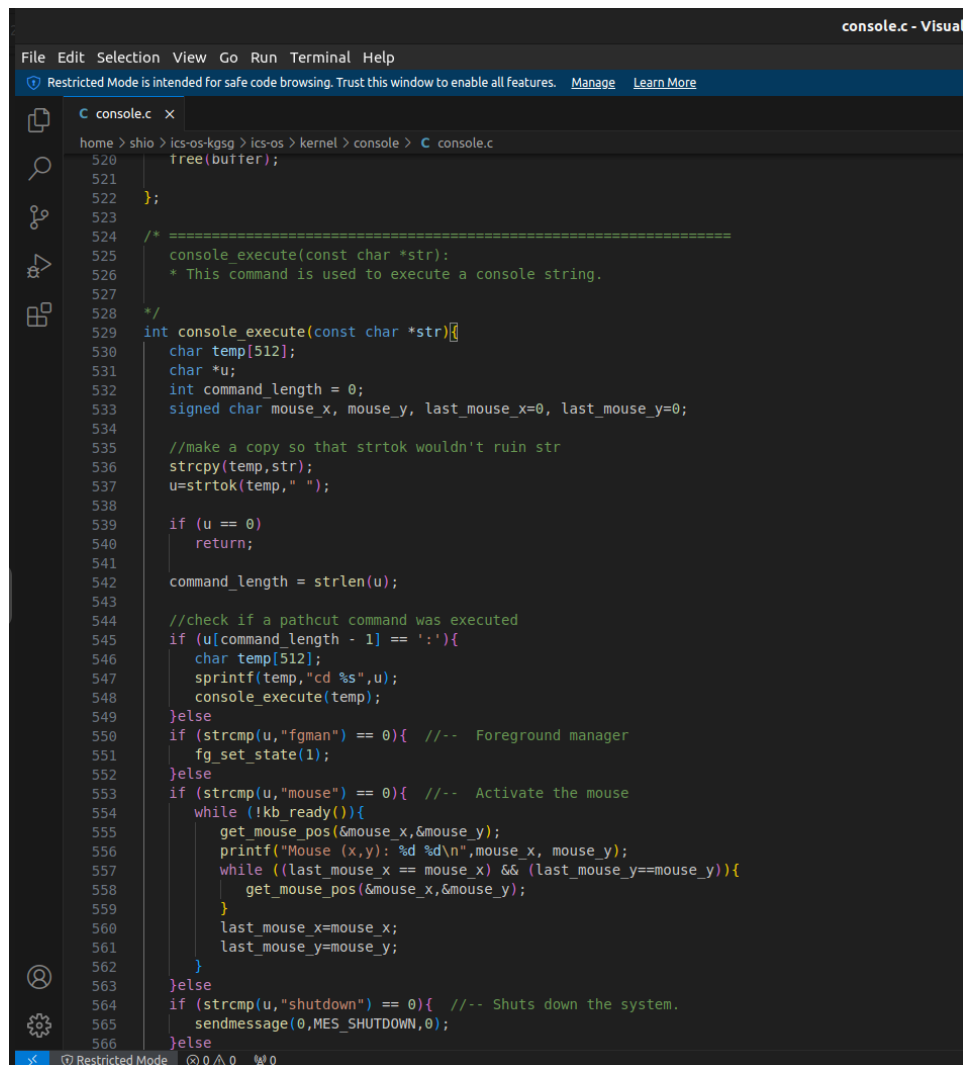
Prerequisites:



```
shio@ksgabinete: ~/ics-os-ksgsg/ics-os/contrib/chown
shio@ksgabinete: ~/ics-os-ksgsg/... x shio@ksgabinete: ~/ics-os-ksgsg/... x shio@ksgabinete: ~/ics-os-ksgsg/... x
shio@ksgabinete:~/ics-os-ksgsg/ics-os$ git branch
lab01
* lab02
master
```

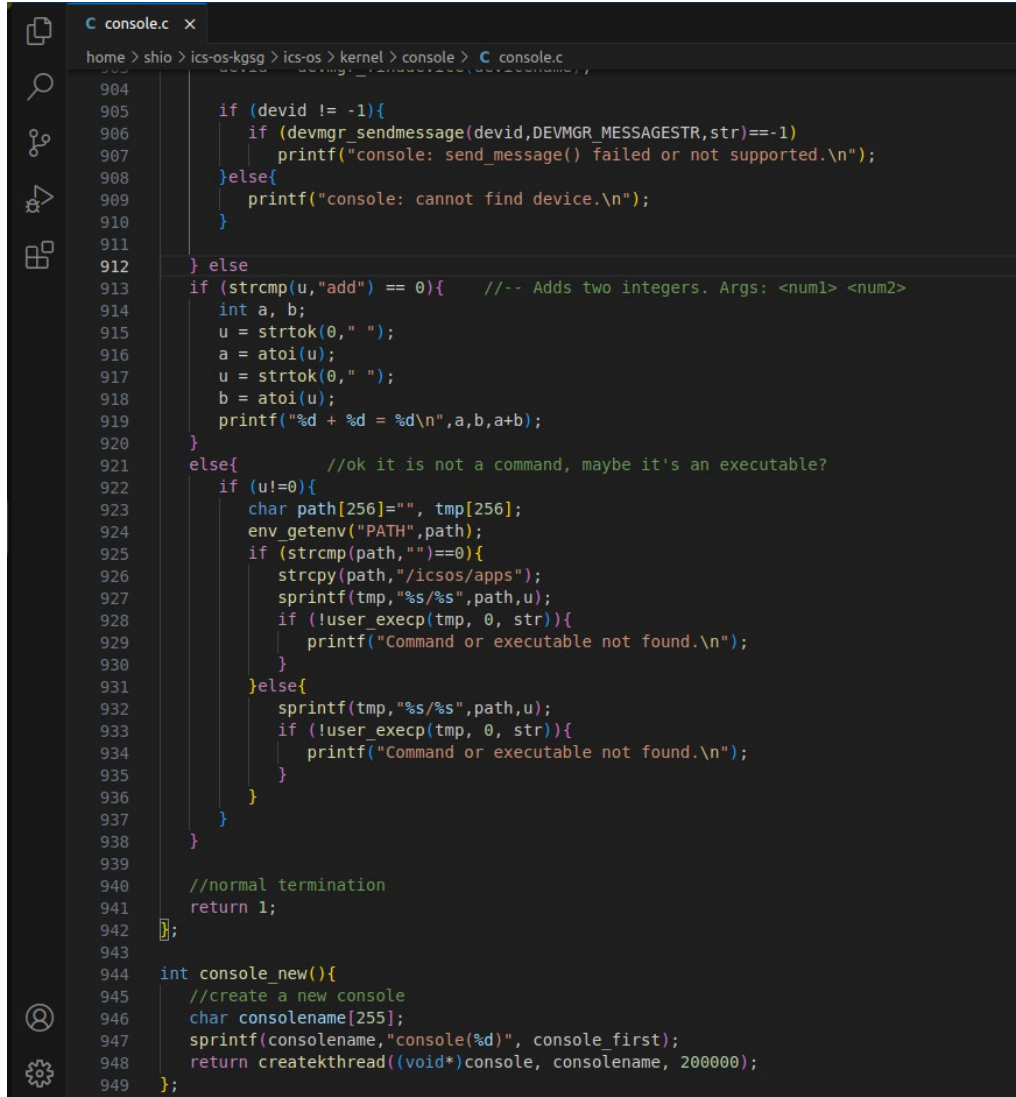
Task 1: Add a new console command (3 points)

I. Opening the implementation file of the Command Line Interface (CLI) in ICS-OS.
(kernel/console/console.c)



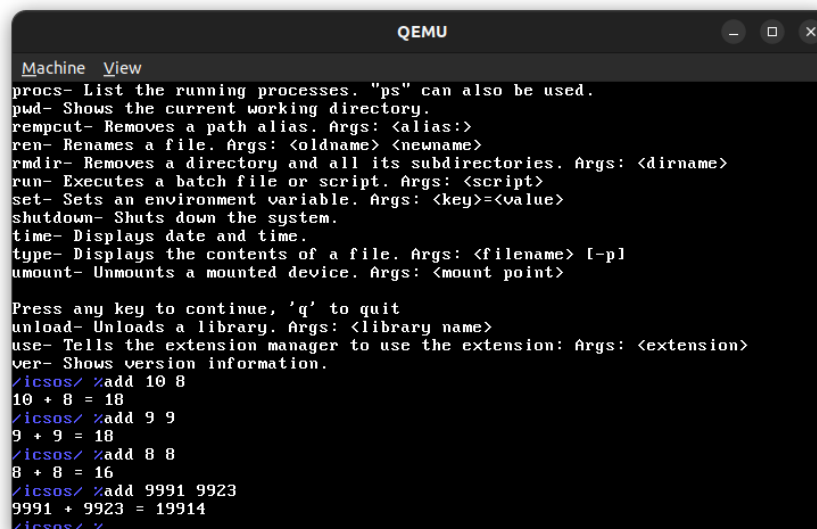
```
console.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
C console.c x
home > shio > ics-os-ksgsg > ics-os > kernel > console > C console.c
520 tree(buffer);
521
522 };
523
524 /* =====
525 console_execute(const char *str):
526 * This command is used to execute a console string.
527 */
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
544     //check if a pathcut command was executed
545     if (u[command_length - 1] == ':'){
546         char temp[512];
547         sprintf(temp,"cd %s",u);
548         console_execute(temp);
549     }else
550     if (strcmp(u,"fgman") == 0){ //-- Foreground manager
551         fg_set_state(1);
552     }else
553     if (strcmp(u,"mouse") == 0){ //-- Activate the mouse
554         while (!kb_ready()){
555             get_mouse_pos(&mouse_x,&mouse_y);
556             printf("Mouse (x,y): %d %d\n",mouse_x, mouse_y);
557             while ((last_mouse_x == mouse_x) && (last_mouse_y==mouse_y)){
558                 get_mouse_pos(&mouse_x,&mouse_y);
559             }
560             last_mouse_x=mouse_x;
561             last_mouse_y=mouse_y;
562         }
563     }else
564     if (strcmp(u,"shutdown") == 0){ //-- Shuts down the system.
565         sendmessage(0,MES_SHUTDOWN,0);
566     }else
567     }
```

II. Editing the “console.c” file to add a code fragment for the new internal command “add” that takes two integer arguments.



```
904
905     if (devid != -1){
906         if (devmgr_sendmessage(devid,DEVVMGR_MESSAGESTR,str)==-1)
907             printf("console: send_message() failed or not supported.\n");
908     }else{
909         printf("console: cannot find device.\n");
910     }
911
912 } else
913 if (strcmp(u,"add") == 0){    //-- Adds two integers. Args: <num1> <num2>
914     int a, b;
915     u = strtok(0, " ");
916     a = atoi(u);
917     u = strtok(0, " ");
918     b = atoi(u);
919     printf("%d + %d = %d\n",a,b,a+b);
920 }
921 else{    //--ok it is not a command, maybe it's an executable?
922     if (u!=0){
923         char path[256]="", tmp[256];
924         env_getenv("PATH",path);
925         if (strcmp(path,"")==0){
926             strcpy(path,"/icsos/apps");
927             sprintf(tmp,"%s/%s",path,u);
928             if (!user_execlp(tmp, 0, str)){
929                 printf("Command or executable not found.\n");
930             }
931         }else{
932             sprintf(tmp,"%s/%s",path,u);
933             if (!user_execlp(tmp, 0, str)){
934                 printf("Command or executable not found.\n");
935             }
936         }
937     }
938 }
939
940 //normal termination
941 return 1;
942 };
943
944 int console_new(){
945     //create a new console
946     char consolename[255];
947     sprintf(consolename,"console(%d)", console_first);
948     return createkthread((void*)console, consolename, 200000);
949 };
```

III. Running the “add” command.



```
QEMU
Machine View
procs- List the running processes. "ps" can also be used.
pwd- Shows the current working directory.
rempcut- Removes a path alias. Args: <alias:>
ren- Renames a file. Args: <oldname> <newname>
rmdir- Removes a directory and all its subdirectories. Args: <dirname>
run- Executes a batch file or script. Args: <script>
set- Sets an environment variable. Args: <key>=<value>
shutdown- Shuts down the system.
time- Displays date and time.
type- Displays the contents of a file. Args: <filename> [-p]
umount- Unmounts a mounted device. Args: <mount point>

Press any key to continue, 'q' to quit
unload- Unloads a library. Args: <library name>
use- Tells the extension manager to use the extension: Args: <extension>
ver- Shows version information.
/icsos/ %add 10 8
10 + 8 = 18
/icsos/ %add 9 9
9 + 9 = 18
/icsos/ %add 8 8
8 + 8 = 16
/icsos/ %add 9991 9923
9991 + 9923 = 19914
/icsos/ %
```

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

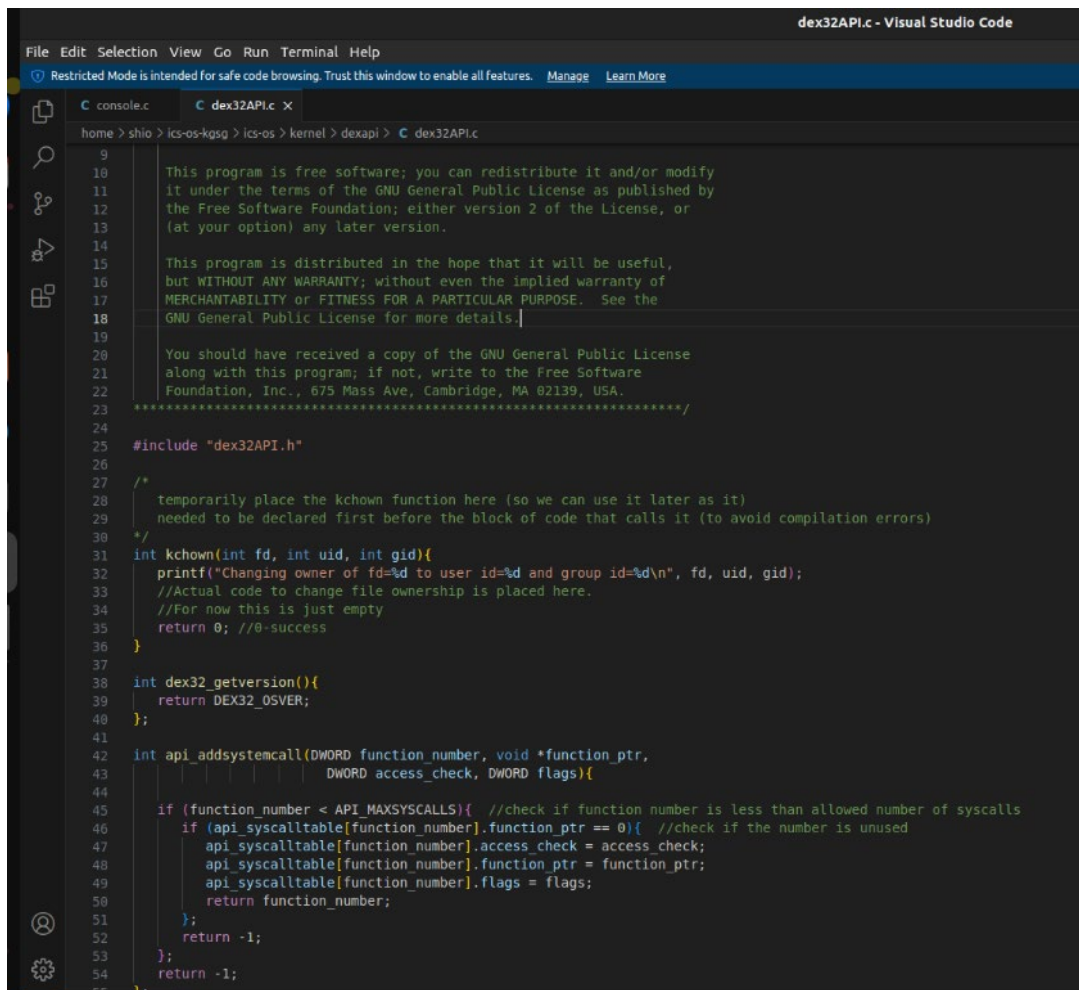
One advantage of having the CLI as part of the kernel itself, instead of a user application like Bash, is flexibility. If you wish to add a new command or fix some bugs in your CLI, you can easily modify the CLI implementation file in the kernel directory. However, this particular advantage can also be one of the disadvantages of having the CLI as part of the kernel itself, instead of a user application like Bash. Why? Since the implementation of the CLI is easily editable, it poses many security concerns, such as allowing anyone to modify some of the code blocks in resource files or completely delete some of these files - both of which can prevent the CLI from behaving correctly once run.

Another advantage I can think of from having the CLI be part of the kernel itself is improved performance. Since it's part of the kernel, then CLI can quickly access the system resources needed for an operation which are also part of the kernel itself.

On the other hand, another disadvantage I can think of is the overly privileged access granted to users. Allowing users to run commands in kernel-level introduces additional security risks to the system.

Task 2: Add a new system call service/function (3 points)

I. Opening and editing the “dex32API.c” file to add a new system call function/service called “kchown()”.



```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C console.c C dex32API.c x
home > shio > lcs-os-kqsg > lcs-os > kernel > dexapi > C dex32API.c

9
10 This program is free software; you can redistribute it and/or modify
11 it under the terms of the GNU General Public License as published by
12 the Free Software Foundation; either version 2 of the License, or
13 (at your option) any later version.
14
15 This program is distributed in the hope that it will be useful,
16 but WITHOUT ANY WARRANTY; without even the implied warranty of
17 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
18 GNU General Public License for more details.
19
20 You should have received a copy of the GNU General Public License
21 along with this program; if not, write to the Free Software
22 Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
23
24
25 #include "dex32API.h"
26
27 /*
28 temporarily place the kchown function here (so we can use it later as it)
29 needed to be declared first before the block of code that calls it (to avoid compilation errors)
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 }
37
38 int dex32_getversion(){
39     return DEX32_OSVER;
40 };
41
42 int api_addsyscall(DWORD function_number, void *function_ptr,
43                     DWORD access_check, DWORD flags){
44
45     if (function_number < API_MAXSYSCALLS){ //check if function number is less than allowed number of syscalls
46         if (api_syscalltable[function_number].function_ptr == 0){ //check if the number is unused
47             api_syscalltable[function_number].access_check = access_check;
48             api_syscalltable[function_number].function_ptr = function_ptr;
49             api_syscalltable[function_number].flags = flags;
50             return function_number;
51         };
52         return -1;
53     };
54     return -1;
55 }
```

II. Adding kchown() to the system call table.

```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

C console.c C dex32API.c x
home > shio > ics-os-kqsg > ics-os > kernel > dexapi > C dex32API.c
164 api_addsystemcall(0x76,Dex32NextLn,0,0);
165 api_addsystemcall(0x77,Dex32PutC,0,0);
166 api_addsystemcall(0x78,Dex32PutChar,0,0);
167 api_addsystemcall(0x79,Dex32FreeDDL,0,0);
168 api_addsystemcall(0x7A,Dex32GetX,0,0);
169 api_addsystemcall(0x7B,Dex32GetY,0,0);
170 api_addsystemcall(0x7C,Dex32SetX,0,0);
171 api_addsystemcall(0x7D,Dex32SetY,0,0);
172 api_addsystemcall(0x7E,Dex32GetAttb,0,0);
173 api_addsystemcall(0x7F,Dex32SetDDL,0,0);
174 api_addsystemcall(0x80,io_setscroll,0,0);
175 api_addsystemcall(0x90,user_fork,0,API_REQUIRE_INTS);
176 api_addsystemcall(0x93,devmgr_ioctl,0,API_REQUIRE_INTS);
177 api_addsystemcall(0x94,devmgr_finddevice,0,0);
178 api_addsystemcall(0x95,console_execute,0,API_REQUIRE_INTS);
179 api_addsystemcall(0x96,getprecisetime,0,API_REQUIRE_INTS);
180 api_addsystemcall(0x97,fcopy,0,API_REQUIRE_INTS);
181 api_addsystemcall(0x98,vfs_listdir,0,API_REQUIRE_INTS);
182 api_addsystemcall(0x99,vfs_mount_device,0,API_REQUIRE_INTS);
183 api_addsystemcall(0x9A,vfs_unmount_device,0,API_REQUIRE_INTS);
184 //----- recently added by jach
185 api_addsystemcall(0x9B,delay,0,API_REQUIRE_INTS);
186 api_addsystemcall(0x9C,kb_ready,0,0);
187 api_addsystemcall(0x9D,write_text,0,0);
188 api_addsystemcall(0x9E,write_char,0,0);
189 api_addsystemcall(0x9F,env_getenv,0,0);
190 api_addsystemcall(0xA0,env_setenv,0,0);
191
192 //----- recently added for lab02 - task02
193 api_addsystemcall(0xC2,kchown,0,0);
194 };
195
196
197 DWORD api_syscall(DWORD fxn,DWORD val,DWORD val2,
198                 DWORD val3,DWORD val4,DWORD val5){
199     char temp[255];
200     DWORD retval = 0; //the return value of a syscall is placed here
201     DWORD (*syscall_function)(DWORD p1,DWORD p2, DWORD p3, DWORD p4, DWORD p5);
202     //cursyscall[] is used for debugging purposes, it stores the last two different
203     //system calls that was called and sets op_success if the system
204     //call finished without crashing or causing a fault, false otherwise
205     if (fxn!=current_process->cursyscall[1]){
206         current_process->cursyscall[0]=current_process->cursyscall[1];
207         current_process->cursyscall[1]=fxn;
208     };
209
210     //place a marker to indicate if a syscall has successfully completed
```

Task 3: Invoke a system call in a system utility (4 points)

Task3a: Create the source

I. Create the chown.exe system utility source code.

```
shio@ksgabinete:~/ics-os-ksgsg/ics-os$ ls
apps      base      docker-compose.yml  ics-os-floppy.img  lib      mnt      sdk
apps-old  boot      Dockerfile          INSTALL            LICENSE  README  TODO
AUTHORS   contrib   grub.img            kernel             Makefile  scripts  vmDEX
shio@ksgabinete:~/ics-os-ksgsg/ics-os$ cd contrib
shio@ksgabinete:~/ics-os-ksgsg/ics-os/contrib$ cp -r hello/ chown/
shio@ksgabinete:~/ics-os-ksgsg/ics-os/contrib$ cd chown
shio@ksgabinete:~/ics-os-ksgsg/ics-os/contrib/chown$ mv hello.c chown.c
shio@ksgabinete:~/ics-os-ksgsg/ics-os/contrib/chown$ sed -i 's/hello/chown/g' Makefile
shio@ksgabinete:~/ics-os-ksgsg/ics-os/contrib/chown$
```

Task3b: Build the executable and install

I. Start the build container to create and install the chown.exe executable.

```
root@e523a150887c: /home/ics-os/contrib/chown
shio@ksgabinete:~/ics-os-ksgsg/ics-os$ docker compose run ics-os-build
root@e523a150887c:/# cd /home/ics-os/contrib/chown/
root@e523a150887c:/home/ics-os/contrib/chown# make
gcc -m32 -w -nostdlib -fno-builtin -static -ochown.exe chown.c ../../sdk/tccsdk.c ../../sdk/libtcc1.c
../../sdk/crt1.c
root@e523a150887c:/home/ics-os/contrib/chown# make install
cp chown.exe ../../apps
root@e523a150887c:/home/ics-os/contrib/chown#
```

Task3c: Run the executable inside ICS-OS

```
QEMU
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
/icsos/apps/ %ls -l -oname
Filename                               Size(bytes)  Attribute   Date Modified
..                                     0            d-rw        11/21/2023
..                                     0            d-rw        11/21/2023
chown.exe                             20216        -xrw        11/21/2023
ed.exe                                53194        -xrw        11/21/2023
hello.exe                             20120        -xrw        11/21/2023
hxdmp.exe                             20236        -xrw        11/21/2023
lzip.exe                              24024        -xrw        11/21/2023
nasm.exe                              308736       -xrw        11/21/2023
pak.exe                               17099        -xrw        11/21/2023
tcc.exe                               186056       -xrw        11/21/2023
vgademo.exe                           20472        -xrw        11/21/2023

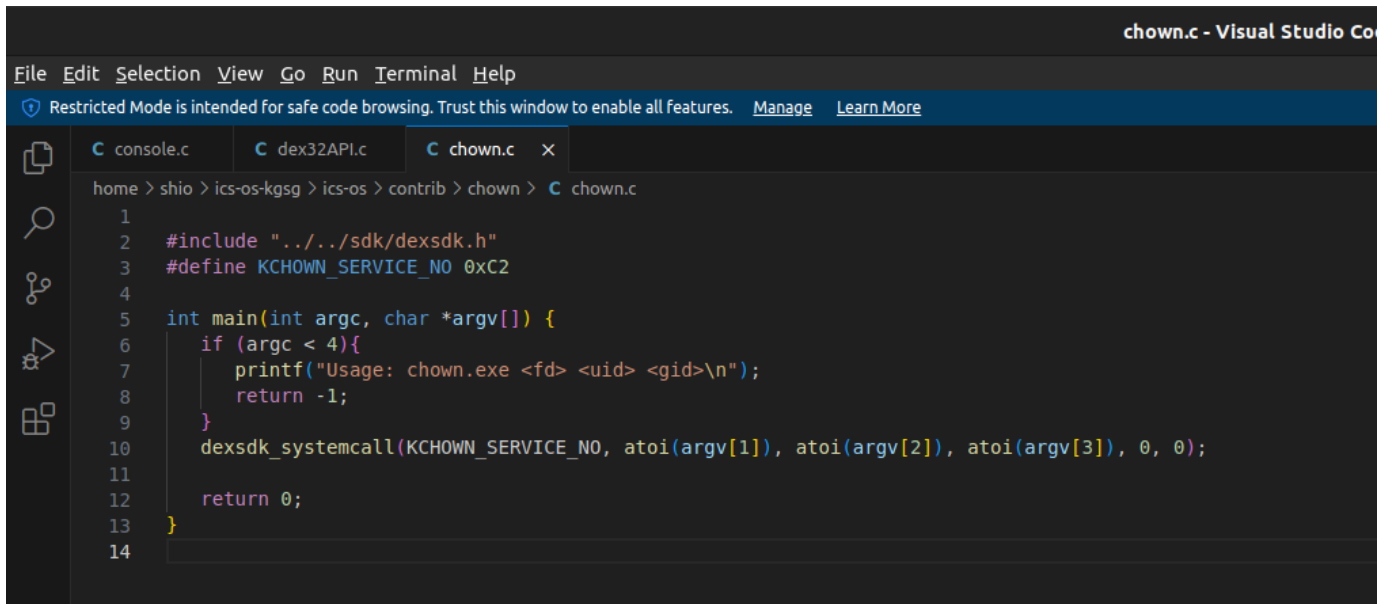
Total Files: 11 Total Size: 670153 bytes
/icsos/apps/ %chown.exe
Hello World from ICS-OS!
/icsos/apps/ %
```

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

"Hello World from ICS-OS!" is the output of executing chown.exe inside ICS-OS.

Task3d: Modify chown.c to invoke the new service via syscall

I. Go back to the contrib/chown folder in the code editing terminal. Edit chown.c and replace its contents.



```
chown.c - Visual Studio Co
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
C console.c C dex32API.c C chown.c X
home > shio > ics-os-ksg > ics-os > contrib > chown > C chown.c
1
2 #include "../../sdk/dextrsdk.h"
3 #define KCHOWN_SERVICE_NO 0xC2
4
5 int main(int argc, char *argv[]) {
6     if (argc < 4){
7         printf("Usage: chown.exe <fd> <uid> <gid>\\n");
8         return -1;
9     }
10    dextrsdk_systemcall(KCHOWN_SERVICE_NO, atoi(argv[1]), atoi(argv[2]), atoi(argv[3]), 0, 0);
11
12    return 0;
13 }
14
```

QUESTION: Study the function dextrsdk_systemcall() defined in sdk/tccsdk.c. What does this function do? Discuss two other functions that call dextrsdk_systemcall().

The function dextrsdk_systemcall() defined in sdk/ tccsdk.c invokes/executes system calls in ICS-OS. It was used/called in other functions like dex_exit() and charputc(). dex_exit() invokes a system call that exits/terminates the ICS-OS. While, charputc() invokes a system call that writes/displays a character in the ICS-OS console/screen.

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

```
total files: 11 total size: 81033 bytes
/icsos/apps/ %chown.exe
Usage: chown.exe <fd> <uid> <gid>
/icsos/apps/ %_

set- Sets an environment variable. Args: <key>=<value>
shutdown- Shuts down the system.
time- Displays date and time.
type- Displays the contents of a file. Args: <filename> [-p]
umount- Unmounts a mounted device. Args: <mount point>
unload- Unloads a library. Args: <library name>
use- Tells the extension manager to use the extension: Args: <extension>
ver- Shows version information.
/icsos/apps/ %chown.exe 01 02 03
Changing owner of fd=1 to user id=2 and group id=3
/icsos/apps/ %chown.exe 8 18 26
Changing owner of fd=8 to user id=18 and group id=26
/icsos/apps/ %_
```

I think chown.exe is a program that imitates the 'chown' command in Linux. In ICS-OS, chown.exe, just like the 'chown' command in Linux, changes the ownership of a file. The argument <fd> stands for the file descriptor, <uid> stands for the user ID of the new owner, and <gid> stands for the group ID of the new group owner. In ICS-OS, when the chown.exe is executed 'chown.c' in '/icsos/contrib/chown/' invokes the system call for kchown which we added in the 'dex32API.c' earlier. For now, the kchown does nothing but prints/displays the passed arguments on the ICS-OS console/screen.

Task 4: Cleanup

I. Exit the build container.

```
QEMU - Press Ctrl+Alt+G to release grab
Machine  View
Warning: 6. Cannot resolve 'Sleep'
Warning: 7. Cannot resolve 'WaitForSingleObject'
Warning: 8. Cannot resolve '__dllonexit'
Warning: 2. Cannot resolve 'abort'
Number of Data directories: 16
Examining export directory..
Obtaining DEX specific entrypoints..
Number of exports = 1
fxn names address = 0xe0016038
fxn addresses     = 0xe0016034
function name: dex32_libmain
Calling dex32 compatible library entrypoint located at e000d420..
DEX Ramdisk Driver 1.03
10000 blocks allocated
Identifying FAT type..
Identifying FAT type..
Using FAT16..
allocating FAT..
writing to FAT..formatting..
done.
Installing Ramdisk as ramdisk
Initialization successful!
Driver was assigned handle 18
Call successful.
```

II. Go back to the master branch of the source code.

```
shio@ksgabinete:~/ics-os-kgsg/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
qemu-system-i386: warning: hub 0 is not connected to host network
shio@ksgabinete:~/ics-os-kgsg/ics-os$ git checkout master
M      ics-os/base/icsos.hlp
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'.
shio@ksgabinete:~/ics-os-kgsg/ics-os$ git branch
  lab01
  lab02
* master
```


Tips

You can use the grep utility to quickly search for strings in files from \$ICSOS_HOME.

Example: `$ grep -rn api_init`

```
shio@ksgabinete:~/ics-os-ksgsg/ics-os$ grep -rn api_init
grep: kernel/Kernel32.sym: binary file matches
kernel/dexapi/dex32API.h:48:void api_init();
kernel/dexapi/dex32API_backup.c:59:void api_init(){
kernel/dexapi/dex32API.c:70:void api_init(){
kernel/mapfile.txt:609: 0x0000000000011827a api_init
grep: kernel/kernel32.o: binary file matches
grep: kernel/Kernel32.bin: binary file matches
kernel/kernel32.c:354: api_init();
shio@ksgabinete:~/ics-os-ksgsg/ics-os$
```

6 Reflection

Write some realizations and questions that crossed your mind while doing this lab.

With the amount of resources I found in the ICS-OS directory, it's fair to say that creating your own operating system is incredibly challenging. Not only does it take a lot of time and effort, but it also requires a lot of fundamental skills and a deep understanding of how an operating system works. If I were to build an OS one day, it's reasonable to assume that I would require a team to accomplish such task, as I couldn't imagine myself building entirely one on my own from scratch.

Few questions that came to mind while doing this lab are: "Why does the chown.exe in ICS-OS takes an integer as file descriptor argument if Linux 'chown' command requires the file name as an argument?"; and, "Is there a way to implement a logic such that when the exit command is entered in the ICS-OS console, the Qemu window will be forced to exit/close as well?"