

ICS-OS Lab 01

Task 1: Install Docker and Docker-Compose

Docker Installation

```
gwy@LAPTOP-3UC7K3M1:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent
   it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

gwy@LAPTOP-3UC7K3M1:~$
```

Docker Compose Installation

```
gwy@LAPTOP-3UC7K3M1:~$ docker compose version
Docker Compose version v2.29.7
gwy@LAPTOP-3UC7K3M1:~$ |
```

Task 2: Clone the repository and explore the source tree

```
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop$ git clone https://github.com/srg-ics-uplb/ics-os.git ics-os-mgat
Cloning into 'ics-os-mgat'...
remote: Enumerating objects: 3557, done.
remote: Counting objects: 100% (408/408), done.
remote: Compressing objects: 100% (195/195), done.
remote: Total 3557 (delta 216), reused 405 (delta 213), pack-reused 3149 (from 1)
Receiving objects: 100% (3557/3557), 26.02 MiB | 634.00 KiB/s, done.
Resolving deltas: 100% (2368/2368), done.
Updating files: 100% (227/227), done.
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop$
```

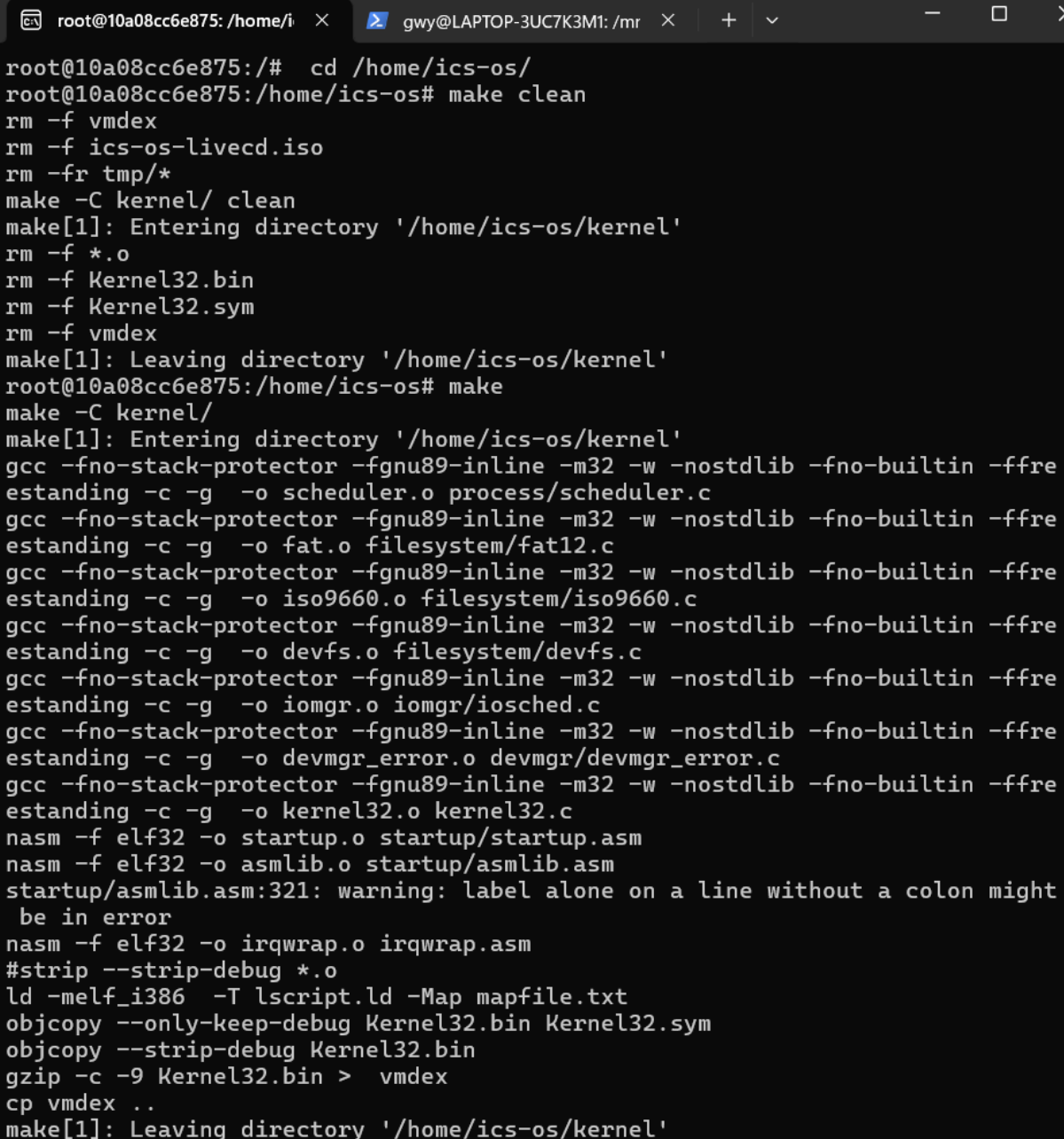
Git Checkout and Branch

```
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop$ cd ics-os-mgat/  
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat$ git checkout -b lab01  
Switched to a new branch 'lab01'  
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat$ git branch  
* lab01  
  master  
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat$
```

Task 3: Build ICS-OS Kernel

```
root@10a08cc6e875: / x gwy@LAPTOP-3UC7K3M1:/mr x + v  
=> [ics-os-build internal] load build definition from Dockerfile  
0.1s  
=> => transferring dockerfile: 328B  
0.1s  
=> [ics-os-build internal] load metadata for docker.io/library/ubuntu:16.04  
4.4s  
=> [ics-os-build internal] load .dockerignore  
0.1s  
=> => transferring context: 2B  
0.0s  
=> [ics-os-build 1/4] FROM docker.io/library/ubuntu:16.04@sha256:1f1a2d56de1d604801a9671f301190704c25d604a416f5 25.7s  
[+] Building 1071.4s (9/9) FINISHED docker:default  
=> [ics-os-build internal] load build definition from Dockerfile 0.1s6  
=> => transferring dockerfile: 328B 0.1s  
=> [ics-os-build internal] load metadata for docker.io/library/ubuntu:16.04 4.4s.  
=> [ics-os-build internal] load .dockerignore 0.1s  
=> => transferring context: 2B 0.0s  
=> [ics-os-build 1/4] FROM docker.io/library/ubuntu:16.04@sha256:1f1a2d56de1d604801 46.2s  
=> => resolve docker.io/library/ubuntu:16.04@sha256:1f1a2d56de1d604801a9671f30119070 0.0s  
=> => sha256:b6f50765242581c887ff1acc2511fa2d885c52d8fb3ac8c4bba131f 3.36kB / 3.36kB 0.0s  
=> => sha256:58690f9b18fca6469a14da4e212c96849469f9b1be6661d2342a 46.50MB / 46.50MB 39.2s2  
=> => sha256:b51569e7c50720acf6860327847fe342a1afbe148d24c529fb81df105e3 857B / 857B 0.3s  
=> => sha256:da8ef40b9ecabc2679fe2419957220c0272a965c5cf7e0269falaeeb8c5 528B / 528B 0.6s5  
=> => sha256:1f1a2d56de1d604801a9671f301190704c25d604a416f59e03c04f5 1.42kB / 1.42kB 0.0s  
=> => sha256:a3785f78ab8547ae2710c89e627783cfa7ee7824d3468cae6835c9f 1.15kB / 1.15kB 0.0s  
=> => sha256:fb15d46c38dcd1ea0b1990006c3366ecd10c79d374f341687eb2cb23a2c 170B / 170B 1.2s  
=> => extracting sha256:58690f9b18fca6469a14da4e212c96849469f9b1be6661d2342a4bf01774 6.5s  
=> => extracting sha256:b51569e7c50720acf6860327847fe342a1afbe148d24c529fb81df105e3e 0.0s  
=> => extracting sha256:da8ef40b9ecabc2679fe2419957220c0272a965c5cf7e0269falaeeb8c56 0.0s  
=> => extracting sha256:fb15d46c38dcd1ea0b1990006c3366ecd10c79d374f341687eb2cb23a2c8 0.0s  
=> [ics-os-build 2/4] RUN apt-get update 105.2s  
=> [ics-os-build 3/4] RUN apt-get install -y build-essential nasm qemu-kvm tcc git 909.9s  
=> [ics-os-build 4/4] RUN mkdir -p /home/ics-os 0.8s  
=> [ics-os-build] exporting to image 4.3s  
=> => exporting layers 4.2s  
=> => writing image sha256:8c88473d1c3bdfdc6c7f487abc30d8e229a70045f0d4decdd4a50354 0.0s  
=> => naming to docker.io/library/ics-os-ics-os-build 0.0s  
=> [ics-os-build] resolving provenance for metadata file 0.1s  
root@10a08cc6e875:/#
```

Root shell inside the container



A terminal window with a dark background and light-colored text. The window title bar shows two tabs: 'root@10a08cc6e875: /home/i' and 'gwy@LAPTOP-3UC7K3M1: /mr'. The terminal content shows a series of commands and their outputs. The user is in a root shell at the prompt 'root@10a08cc6e875:/#'. They navigate to '/home/ics-os/' and run 'make clean', which removes various files. Then they run 'make', which compiles the kernel. The compilation process involves gcc for C files and nasm for assembly files. A warning is shown for 'startup/asmlib.asm:321: warning: label alone on a line without a colon might be in error'. Finally, the kernel is stripped and compressed into 'vmdex'.

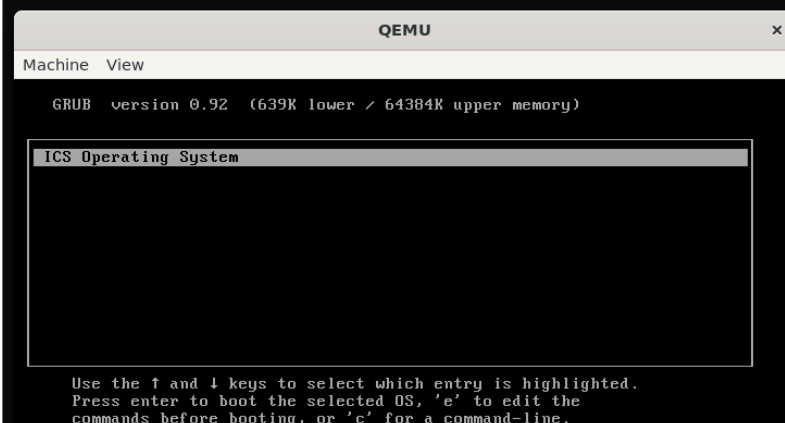
```
root@10a08cc6e875:/# cd /home/ics-os/
root@10a08cc6e875:/home/ics-os# make clean
rm -f vmdex
rm -f ics-os-livcd.iso
rm -fr tmp/*
make -C kernel/ clean
make[1]: Entering directory '/home/ics-os/kernel'
rm -f *.o
rm -f Kernel32.bin
rm -f Kernel32.sym
rm -f vmdex
make[1]: Leaving directory '/home/ics-os/kernel'
root@10a08cc6e875:/home/ics-os# make
make -C kernel/
make[1]: Entering directory '/home/ics-os/kernel'
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o scheduler.o process/scheduler.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o fat.o filesystem/fat12.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o iso9660.o filesystem/iso9660.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o devfs.o filesystem/devfs.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o iomgr.o iomgr/iosched.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -c -g -o devmgr_error.o devmgr/devmgr_error.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffre
standing -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 ..
make[1]: Leaving directory '/home/ics-os/kernel'
```

Task 4: Create the disk and boot ICS-OS

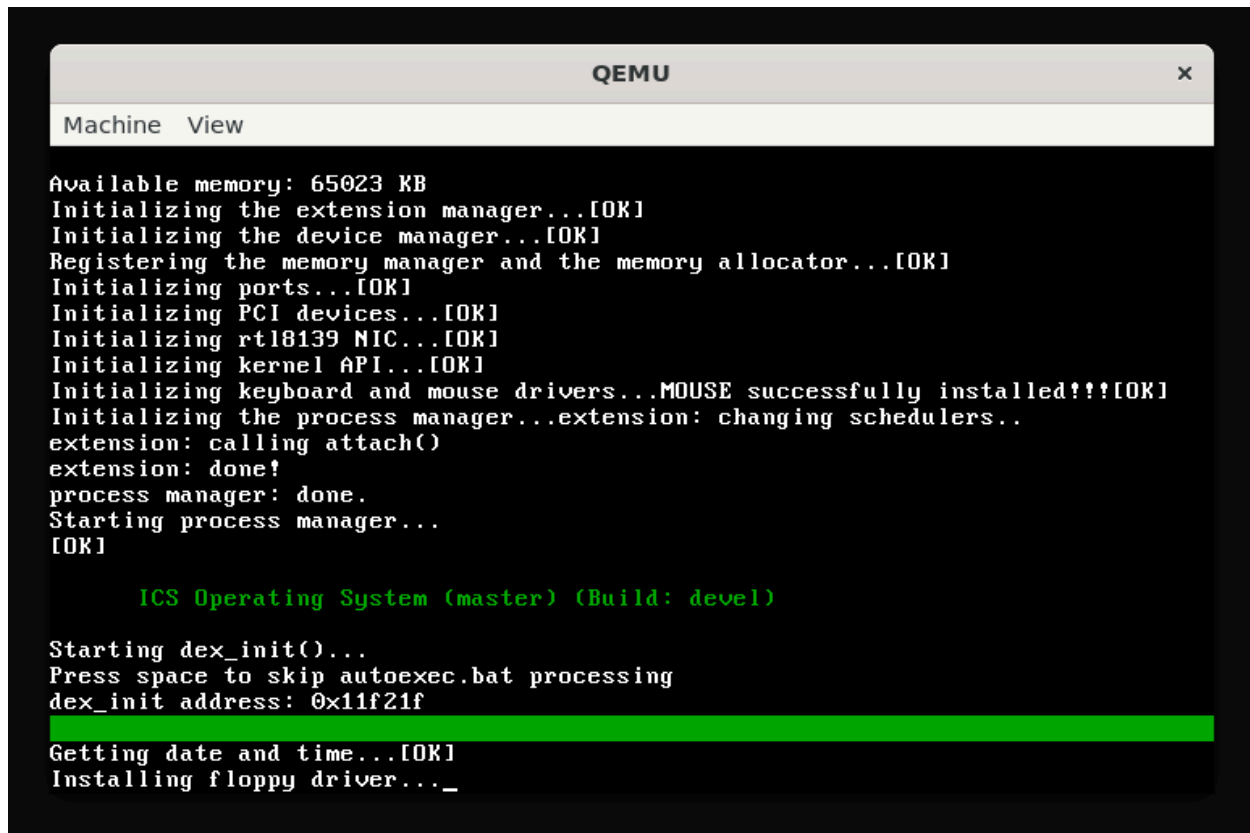
```
root@8b45afd5f25a:/home/ics-os# exit
exit
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ sudo make floppy
rm -fr tmp
mkdir tmp
cp -r vmdex tmp
scripts/gen-help.sh
cp base/* tmp
mkdir -p tmp/apps
mkdir -p tmp/tcc1
mkdir -p tmp/lib1
cp apps/* tmp/apps/
cp sdk/* tmp/tcc1/
cp lib/* tmp/lib1/
cp grub.img ics-os-floppy.img          #copy an image with grub
sudo rm -fr mnt
sudo mkdir mnt
sudo mount ics-os-floppy.img mnt -tmsdos -o loop
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$ |
```

GRUB boot menu

```
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
```



mg' and probing guessed raw.
raw images, write operations on
estrictions.
lease set a backend using '-mach
twork




```
QEMU
Machine View

Available memory: 65023 KB
Initializing the extension manager...[OK]
Initializing the device manager...[OK]
Registering the memory manager and the memory allocator...[OK]
Initializing ports...[OK]
Initializing PCI devices...[OK]
Initializing rtl8139 NIC...[OK]
Initializing kernel API...[OK]
Initializing keyboard and mouse drivers...MOUSE successfully installed!!![OK]
Initializing the process manager...extension: changing schedulers..
extension: calling attach()
extension: done!
process manager: done.
Starting process manager...
[OK]

    ICS Operating System (master) (Build: devel)

Starting dex_init()...
Press space to skip autoexec.bat processing
dex_init address: 0x11f21f
Getting date and time...[OK]
Installing floppy driver..._
```



```
QEMU
Machine View

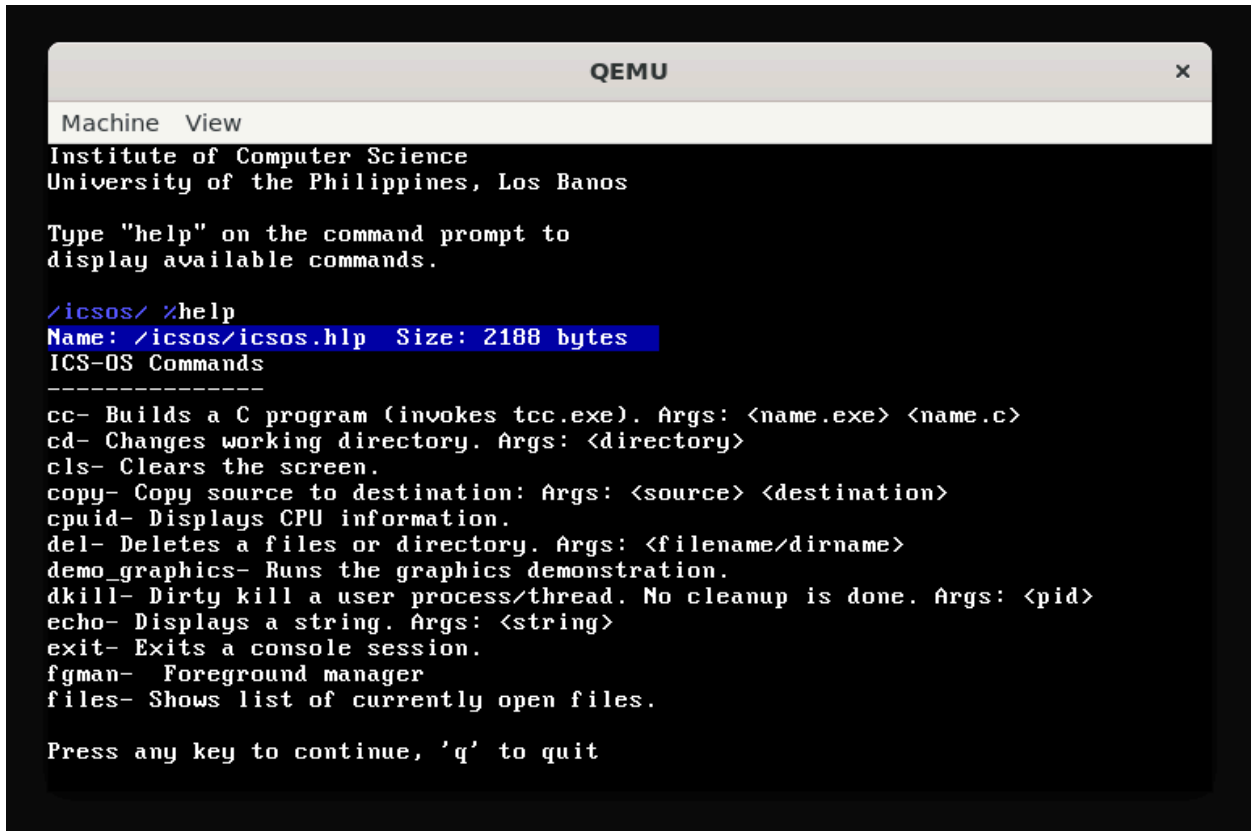
Welcome to the ICS Operating System

Institute of Computer Science
University of the Philippines, Los Banos

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

/icsos/ %
```

Task 5: Run ICS-OS commands



The screenshot shows a QEMU window titled "QEMU" with a close button. Inside the window, the text "Machine View" is at the top. Below it, the text "Institute of Computer Science" and "University of the Philippines, Los Banos" is displayed. A message says "Type 'help' on the command prompt to display available commands." The command prompt shows the user has entered "/icsos/ %help". The output is a list of ICS-OS commands and their descriptions, including cc, cd, cls, copy, cpuid, del, demo_graphics, dkill, echo, exit, fgman, and files. The prompt "Press any key to continue, 'q' to quit" is at the bottom.

```
QEMU
Machine View
Institute of Computer Science
University of the Philippines, Los Banos

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

/icsos/ %help
Name: /icsos/icsos.hlp Size: 2188 bytes
ICS-OS Commands
-----
cc- Builds a C program (invokes tcc.exe). Args: <name.exe> <name.c>
cd- Changes working directory. Args: <directory>
cls- Clears the screen.
copy- Copy source to destination: Args: <source> <destination>
cpuid- Displays CPU information.
del- Deletes a files or directory. Args: <filename/dirname>
demo_graphics- Runs the graphics demonstration.
dkill- Dirty kill a user process/thread. No cleanup is done. Args: <pid>
echo- Displays a string. Args: <string>
exit- Exits a console session.
fgman- Foreground manager
files- Shows list of currently open files.

Press any key to continue, 'q' to quit
```

Run two commands

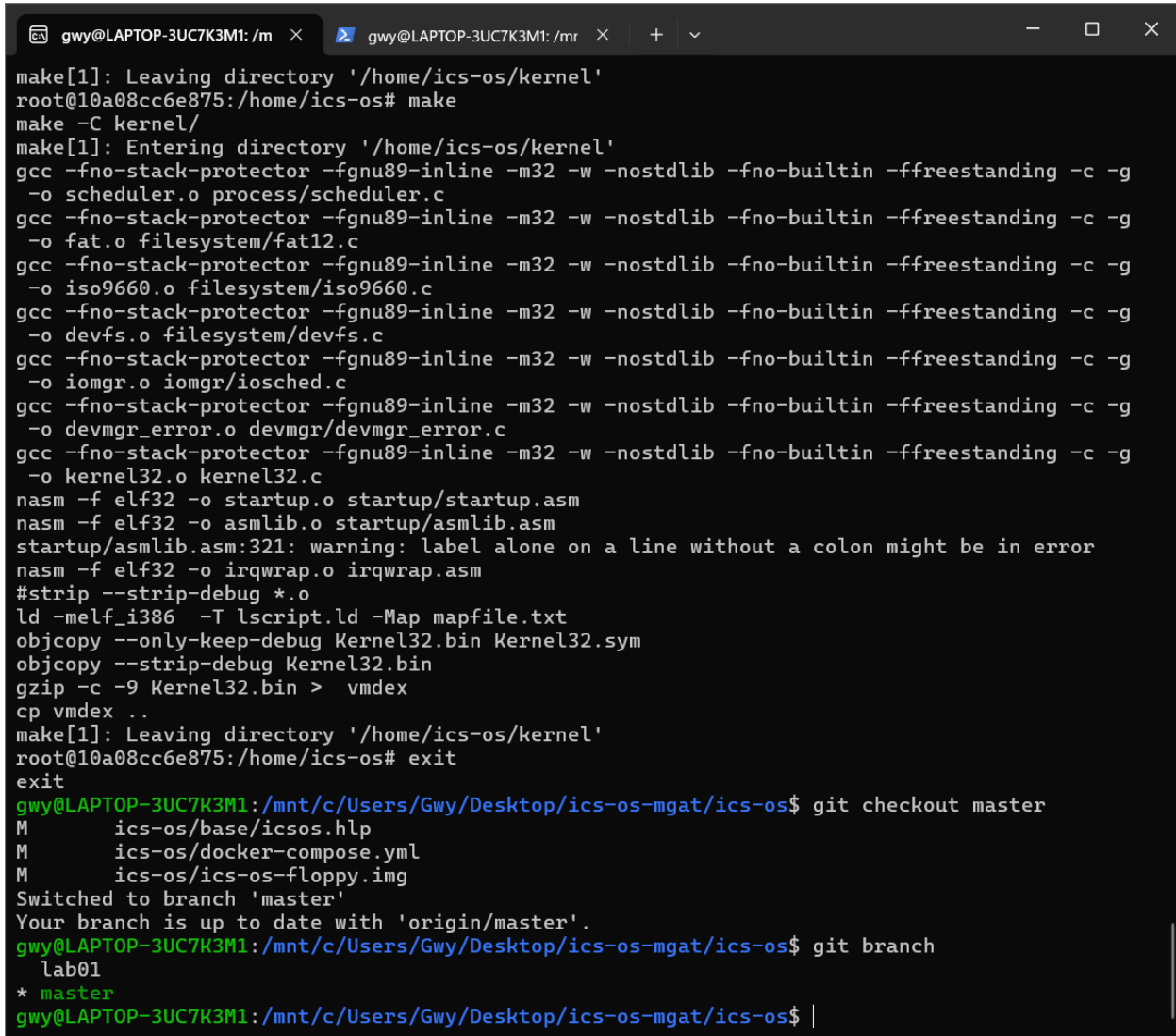


The first screenshot shows the command prompt with the user entering "/icsos/ %ls". The output lists files and directories: apps, boot, lib1, tcc1, autoexec.bat, icsos.hlp, icsoshlp.txt, license, and vmdex. It also shows "Total Files: 9" and "Total Size: 117033 bytes". The second screenshot shows the command prompt with the user entering "/icsos/ %ver". The output displays "ICS Operating System (master)" and "APM information: CS32 base:0 CS16 base:0 DS base:0 offset:0".

```
/icsos/ %ls
apps          boot          lib1
tcc1          autoexec.bat icsos.hlp
icsoshlp.txt  license      vmdex
Total Files: 9 Total Size: 117033 bytes
/icsos/ %

/icsos/ %ver
ICS Operating System
(master)
/icsos/ %off
APM information: CS32 base:0 CS16 base:0 DS base:0 offset:0
```

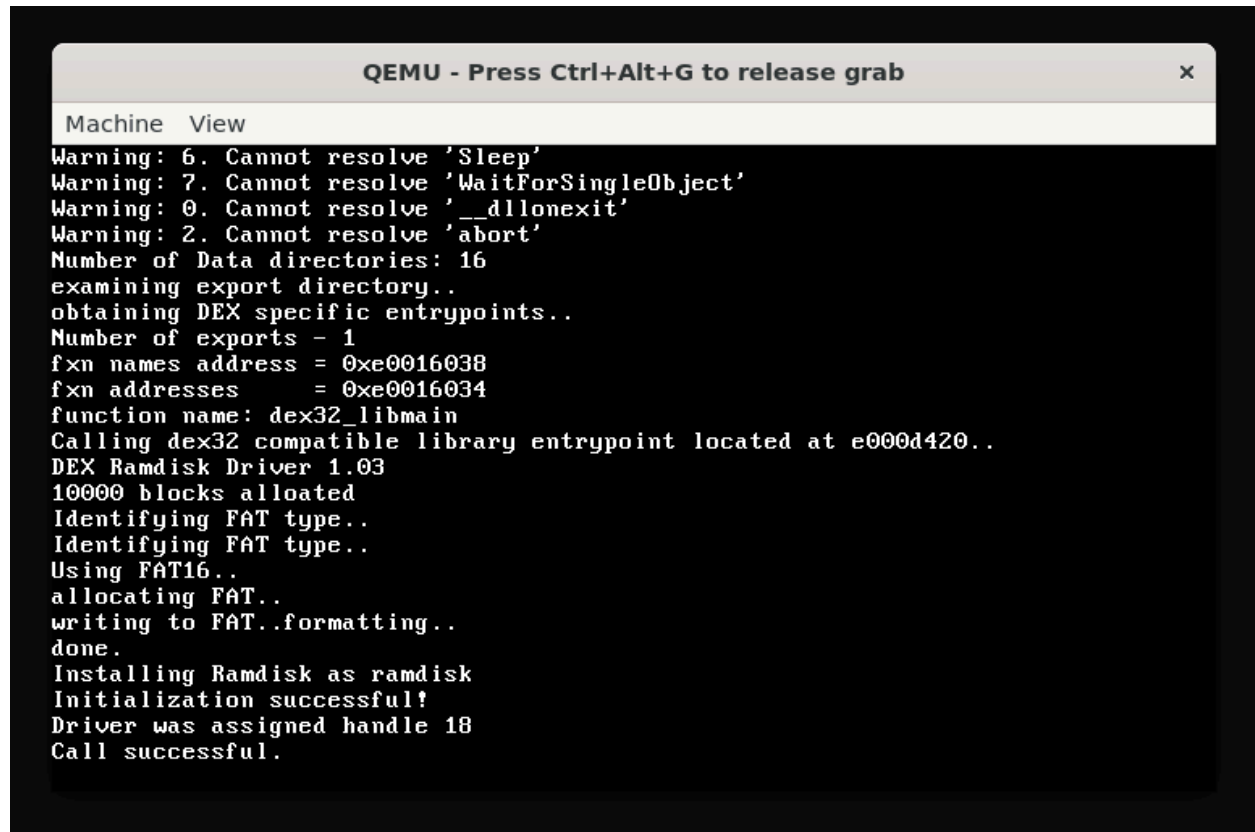
Task 6: Cleanup



```
gwy@LAPTOP-3UC7K3M1: /m
make[1]: Leaving directory '/home/ics-os/kernel'
root@10a08cc6e875:/home/ics-os# make
make -C kernel/
make[1]: Entering directory '/home/ics-os/kernel'
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g
-o scheduler.o process/scheduler.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g
-o fat.o filesystem/fat12.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g
-o iso9660.o filesystem/iso9660.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g
-o devfs.o filesystem/devfs.c
gcc -fno-stack-protector -fgnu89-inline -m32 -w -nostdlib -fno-builtin -ffreestanding -c -g
-o iomgr.o iomgr/iosched.c
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 ..
make[1]: Leaving directory '/home/ics-os/kernel'
root@10a08cc6e875:/home/ics-os# 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
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$ git branch
lab01
* master
gwy@LAPTOP-3UC7K3M1:/mnt/c/Users/Gwy/Desktop/ics-os-mgat/ics-os$ |
```

REFLECTION

First, I installed the needed packages and entered all the necessary commands in the terminal. I was able to boot the ICS OS successfully. I was amazed and thought of the amount of time it took to work on the OS itself. Also, I realized that the process is the same as the one we did in the exercise in bootloading, although the OS there is much simpler. One question that crossed my mind was why the OS did not terminate when I typed “exit.” I needed to exit using the close tab. I captured a screenshot and attached it below.



The image shows a screenshot of a QEMU terminal window. The window title is "QEMU - Press Ctrl+Alt+G to release grab" with a close button (X) on the right. Below the title bar, there is a tab labeled "Machine View". The terminal displays the following text:

```
Warning: 6. Cannot resolve 'Sleep'
Warning: 7. Cannot resolve 'WaitForSingleObject'
Warning: 0. 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.
```