

# CSE303 FUNDAMENTALS of OPERATING SYSTEMS

## The MINIX OS

# History of MINIX

- In 1975 the source for UNIX version 6 was available from AT&T
  - Universities took advantage of this and used the source in courses on OS design and implementation.
- With the release of UNIX version 7, AT&T decided to restrict the use of the source.

# History of MINIX (Cont.)

- Dr. Tanenbaum wrote MINIX (from scratch) as a replacement for AT&T UNIX in OS design and implementation courses.
  - Originally, MINIX was written to be compatible with UNIX V7.
  - Since, it has become increasingly POSIX compliant.

# MINIX and Linux

- Independent users of Minix were unsatisfied with its limited capabilities.
  - Linus Torvalds decided to write his own OS based on Minix.

# MINIX

- Minix – Mini Unix (Minix) basically, a UNIX -compatible operating system.
- Minix is small in size, with microkernel-based design.
- Minix has been kept (relatively) small and simple.
- Minix is small, it is nevertheless a preemptive, multitasking operating system.
- Internally, Minix is highly modular, consisting of several system processes organized into a number of independent programs.

# MINIX Facts

- The main MINIX OS is written in C language, in the MINIX2.0 source code, with ".C" at the end of the file is C language source program
- MINIX kernel source code is a small portion of assembler language, most of which are about the underlying program interrupt and exception handling. In the MINIX2.0 source code, with ".S" at the end of the file as the assembly language source program.

# Directory Structure

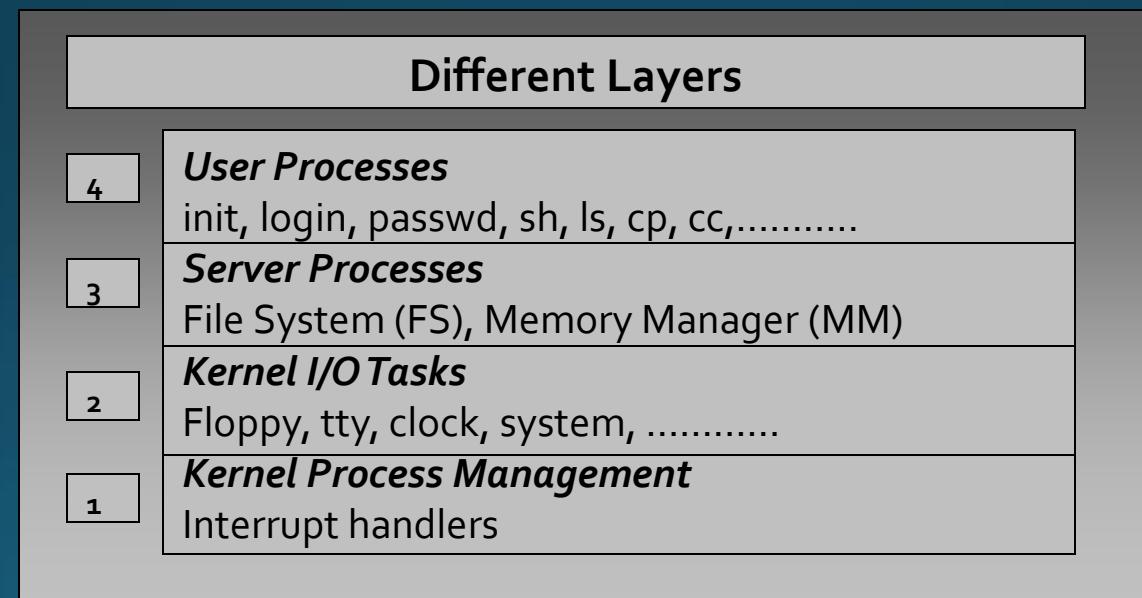
- **/usr/src/kernel/** The directory for MINIX operating system the first layer (process management) and second layer (Facility Management) of the source code,
- **/usr/src/mm/** Catalog stored memory management subsystem of the MINIX operating system in the third layer of the source code,
- **/usr/src/fs/** The FS directory is stored with the file management system of MINIX operating system in the third layer of the source code,
- **/usr/src/lib** directory using the MINIX operating system library routines source code(e.g. open, read),
- **/usr/src/tools** Directory, a lInit source program, used to start Minix.

# Architecture (Cont.)

- There are 4 layers in the Architecture
- Layers 1 to 3 comprise the Minix operating system, with applications running in Layer 4.
- The operating system code is linked **into three totally separate programs** **mm** (the memory manager), **fs** (the file system), and **kernel** (layers 1 and 2).
- Processes in layer 2 and interrupt handlers in layer 1 can communicate by **message passing or shared memory**; **all other communication is through message passing**.

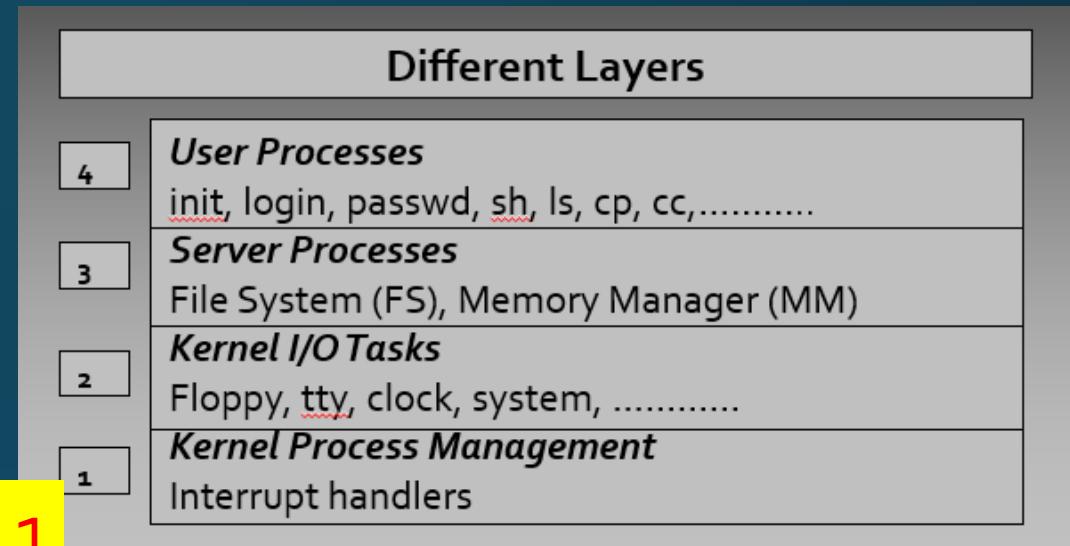
# Internal Architecture

- The figure represents the internal architecture of Minix.



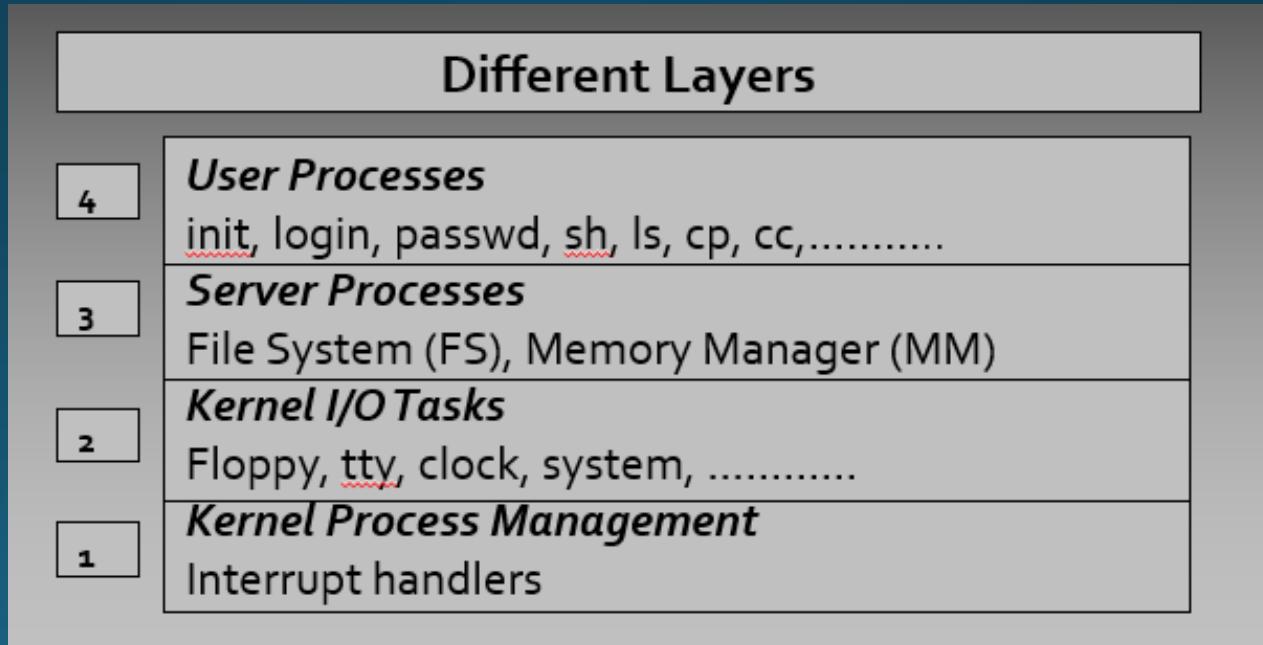
# Detailed Analysis of the Different Layers.

- **Layer 1(Kernel Process Management)** implements communicating sequential processes, thus allowing the remainder of Minix to be implemented as a set of communicating sequential processes. **Layer 1** is a group of interrupt handlers, with its own stack space within kernel.
- **Layer 2 (Kernel I/O Tasks)** contains **device driver processes**. System provides an interface between kernel and the layer 3 processes.
- **Layer 3 (Server Processes)** contains the memory manager and the file system.



# Detailed Analysis of the Different Layers (Cont.)

- Layer 4 (**User Processes**) contains a Unix-like process hierarchy rooted at process 1. Every system call<sup>"</sup> made by a user process in layer 4 is converted into a message to one or other of these processes.

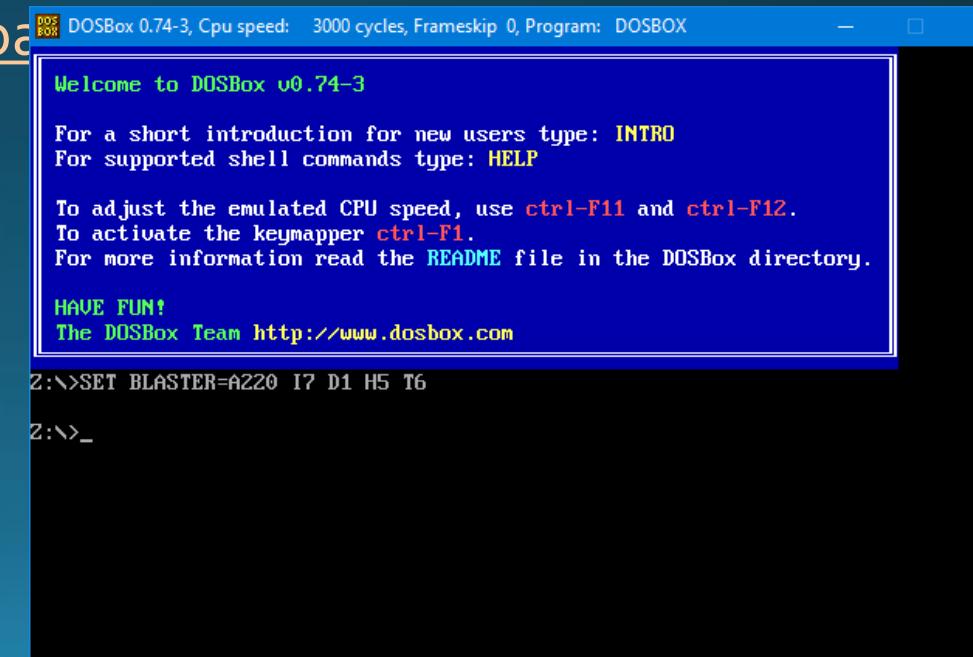


# Unix Basics

Command	Description	Example
ps	Process status (list)	ps -u \$USER
kill	Terminate process	kill -9 3291
pwd	Print working directory	pwd
mv	Move (or rename) file/directory	mv p1.c project1.c
cp	Copy file	cp p1.c p1.bak
more	Display text file a page at a time	more project1.c
pico	Simple text editor	pico +20 project1.c
emacs	Advanced (graphical) text editor	emacs project1.c
diff	Compare files	diff p1.c project1.c

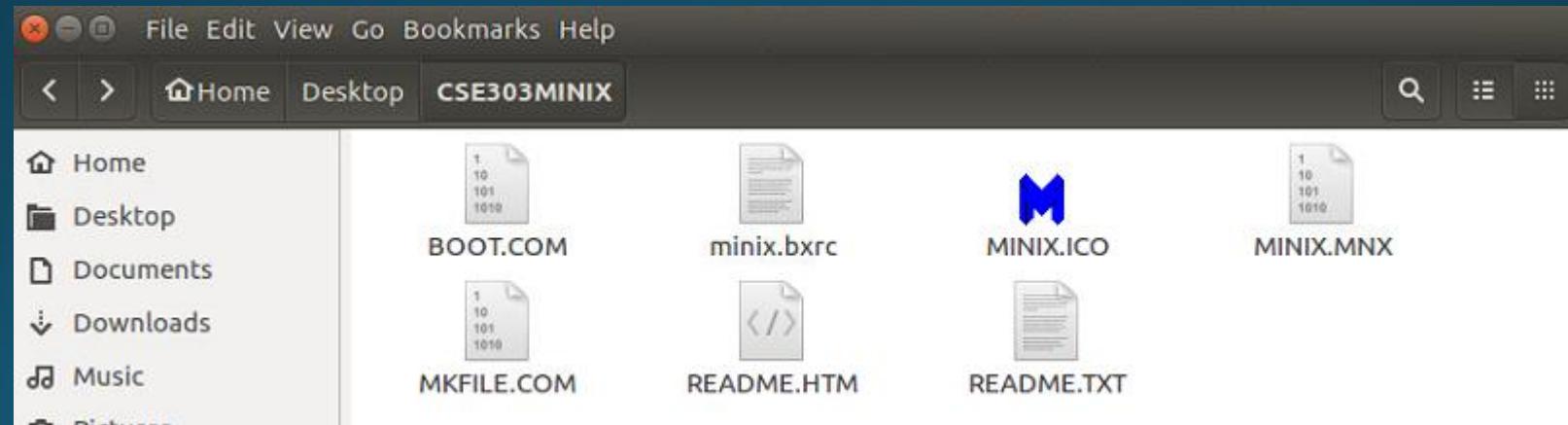
# Required Software

- The MINIX 2.0.4
  - <https://minix1.woodhull.com/current/2.0.4/i386/DOSMINIX.ZIP>
- Installing Dosbox on Windows OS
  - Option 1 - https://www.dosbox.com/downloads
  - Option 2 - WSL
    - Win 11 and dedicated graphics drivers required
- Installing on a Linux – e.g.Ubuntu
  - sudo apt-get install dosbox



# Installation Steps

- Install Ubuntu either as a virtual OS or a regular OS
- Install DOSBOX
  - `#sudo apt-get install dosbox`
- Extract the DOSMINIX.zip file to a folder CSE303MINIX.



# Start the dosbox

- #dosbox
- You can also change the keyboard layout using
  - #keyb tr179
- Initially there will be no C: drive, therefore you must mount the C drive using the command
  - #mount c: /home/[username]/Desktop/CSE303MINIX

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSE

```
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:>SET BLASTER=A220 I7 D1 H5 T6
Z:>_
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
Welcome to DOSBox v0.74

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:>SET BLASTER=A220 I7 D1 H5 T6
Z:>keyb tr179
Keyboard layout tr179 loaded for codepage 857
Z:>mount c: /home/taner/Desktop/CSE303MINIX
Drive C is mounted as local directory /home/taner/Desktop/CSE303MINIX/
Z:>_
```

# Accessing MINIX

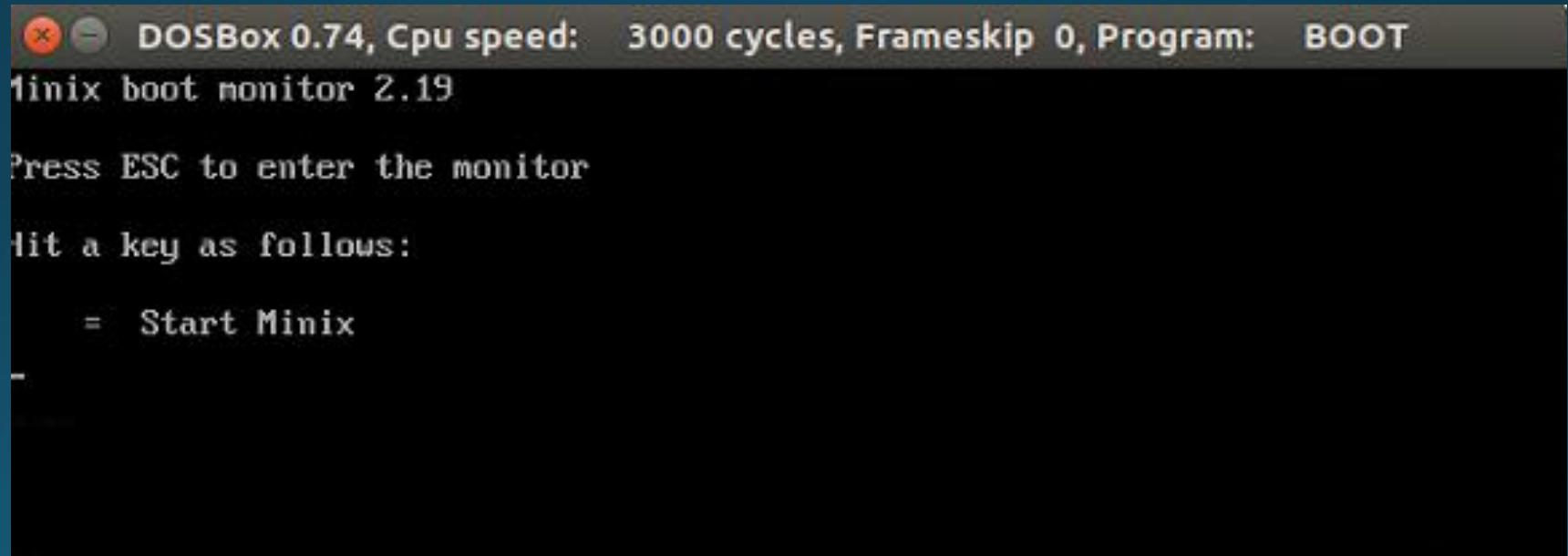
- Change the drive to the C: drive and list the contents of the drive

The screenshot shows a DOSBox window with the following command-line session:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>keyb tr179
Keyboard layout tr179 loaded for codepage 857
Z:\>mount c: /home/taner/Desktop/CSE303MINIX
Drive C is mounted as local directory /home/taner/Desktop/CSE303MINIX/
Z:\>c:
C:\>dir
Directory of C:\>
.
..
BOOT    COM        25-09-2017 23:44
MINIX   ICO        25-09-2017 23:44
MINIX   MNX      20,800 14-09-2003 15:02
MINIX~1 BXR        766 20-12-1998 17:06
MRFILE  COM        52,428,800 09-11-2003 18:47
README  HTM        481 05-04-2003 13:22
README  TXT        2,848 14-09-2003 15:02
 README  TXT        1,650 22-05-2001 23:31
 README  TXT        1,327 22-05-2001 23:31
 7 File(s)      52,456,672 Bytes.
 2 Dir(s)       262,111,744 Bytes free.
C:\>
```

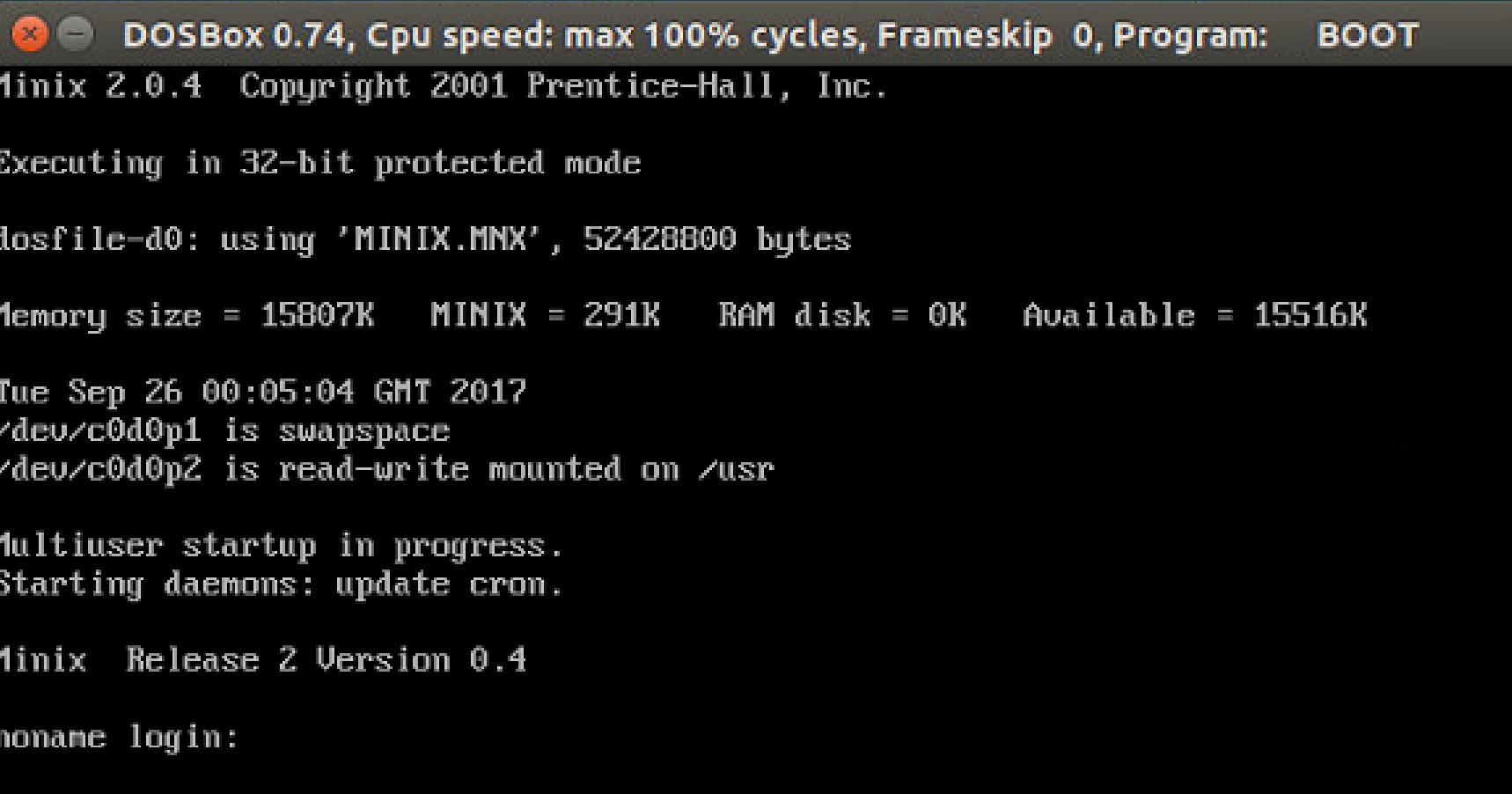
# Booting MINIX

- Enter the boot command as follows
  - # boot MINIX.MNX
  - After the MINIX machine boots, it enters the MINIX boot monitor. The boot monitor screen appears as follows:



# Booting Minix (Cont.)

- When the user hits = key, the following screen appears



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BOOT  
Minix 2.0.4 Copyright 2001 Prentice-Hall, Inc.  
Executing in 32-bit protected mode  
dosfile-d0: using 'MINIX.MNX', 52428800 bytes  
Memory size = 15807K MINIX = 291K RAM disk = 0K Available = 15516K  
Tue Sep 26 00:05:04 GMT 2017  
/dev/c0d0p1 is swap space  
/dev/c0d0p2 is read-write mounted on /usr  
Multiuser startup in progress.  
Starting daemons: update cron.  
Minix Release 2 Version 0.4  
noname login:

# Logging in

- The default username for is **root** without any password.

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: BOOT
Minix 2.0.4 Copyright 2001 Prentice-Hall, Inc.

Executing in 32-bit protected mode

dosfile-d0: using 'MINIX.MNX', 52428800 bytes

Memory size = 15807K MINIX = 291K RAM disk = 0K Available = 15516K

Tue Sep 26 00:05:04 GMT 2017
/dev/c0d0p1 is swapspace
/dev/c0d0p2 is read-write mounted on /usr

Multiuser startup in progress.
Starting daemons: update cron.

Minix Release 2 Version 0.4

noname login: root
#
noname login: root
# ls
.ashrc .exrc bin dev fd0 minix root usr
.ellepro.b1 .profile boot etc fd1 mnt tmp
#
```

# Log out

- #shutdown
- #exit

# Development

- MINIX has built in C compiler named cc
- Editors: Minix provides several editors to edit files. You can get more information using the manual pages of the Minix
  - vi (man vi)
  - elle
  - mined

# Compiling the kernel

- The image of the MINIX kernel is stored in the **/minix** directory in the **test\_kernel** partition. This is where compiled versions of the kernel will appear. At boot, the machine will use the latest version of the kernel from that directory.
- when building the kernel, the new kernel image is designated with a new release number.
  - minix
  - minix20r0
  - minix20r1

# Compiling the kernel (Cont.)

- The Kernel source
  - The MINIX source is located in **/usr/src**
  - Under its various subdirectories, you will find the source code (in C) and various utilities.
  - The source is described in Tanenbaum and Woodhull's *Operating Systems: Design and Implementation*.
  - The files are probably owned by **root**; in order to change the source and rebuild the kernel, you should be **logged in as root** as well.

# Compiling the kernel (Cont.)

- After modifying the kernel source, you must rebuild the kernel.
- This is done by using the **make** utility included in the MINIX distribution.
- Under the **/usr/src/tools** directory are all the files needed to build the kernel.
- In order to build the kernel, you must **cd** to this directory.  
Afterwards, follow these steps:
  - **Recompile the source.** To do this, type "**make**" at the prompt.
  - **Build the new kernel.** Though the new kernel has been compiled, a new kernel image has yet to be built. You can install the kernel on the hard disk by typing type "**make hdboot**".
  - **reboot**

# Compiling the kernel (Cont.)

- `#cd /usr/src/tools`
- `#make hdboot`
- `#reboot`
- After the `make hdboot` command, you will find the new kernel at
  - `/minix/2.0.4rX` where `X` is the version of the new kernel
    - Original is 2.0.4, 1st new is 2.0.4r0, second 2.0.4r1 etc...
- During the boot process minix selects the most up-to-date kernel from the `/minix` directory

# How to recover from the crash?

- Note that the system puts the new compiled image on /minix folder
  - Therefore, before compiling the new image file if you backup the old image file you can protect yourself from the kernel crash errors.
    - `#cp /minix/2.0.4r2 /minix/2.0.4r2.bak`
    - `#cd /usr/src/tools`
    - `#make hdboot`
    - `#reboot`
  - **In order to fix a kernel crash**, where the system does not boot correctly
    - Reboot Minix and hit the ESC key during the bootup process.
    - Set the image file as the backup file
      - `image=/minix/2.0.4r2.bak`
      - `boot`
    - After that you will be able to use the working kernel to fix the problems

# Useful tips

- vi
  - :set number
  - :wq
- Pico like editor : **elle**
  - If you dont know how to use it then type
    - #man elle
- Search for file: **find . -name idle.c**
- Use **grep**

# Write a test application under Minix

- Write a test application under Minix just to print Hello World text.
- You can use any of the editors under Minix