Overview of Architecture and Application of Clone OS

Divyavrat Jugtawat
divyavrat@live.com
JIET, Jodhpur, India

Abstract

Clone OS was made to be a portable OS. It can run from any drive with portable usage. Clone OS en-capsules many new ideas we had for a controlled OS. We are going to explain these new features along with the new freedom it provides to its users. Books can be read on the move, on any computer using this OS. Also ASCII art was never truly supported by other OS before. Clone OS has inbuilt ASCII editors with complete saving and creating frames with each ASCII page. Clone OS also has binary code environment to allow machine coders to code there programs directly with them being saved as a file. This can be used instead of the costly machine kits or simulators. Clone OS also has DOS compatibility, i.e. COM programs of DOS can be run in Clone OS and vice-versa. This extends to all OS that have a DOS API, including Windows till Xp, React OS and CP/M OS. Clone OS allows more and more support for existing materials and so all existing e-book formats like TXT, DOC are supported and others can be converted to these formats. Following is the Description of a new Operating System and the new methods used by it. Its new shell interface and commands. It's API and interrupts are explained.

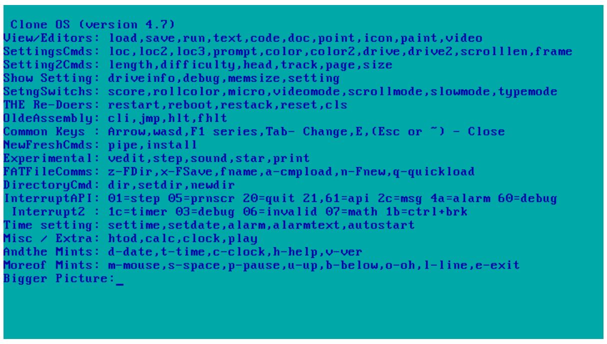
Keywords –

Clone, OS, API, portable OS, USB, command, single-tasking, ASCII art.

Introduction –

There was always a need to create a portable working environment. Most linux distributions allow this but files that are made are not saved once the live CD/USB is removed. Hence we started creating a bootable USB with small OS that can perform basic file management and compatibility with existing OS. Hence Clone OS has ported basic file formats like TXT to work with. Clone OS has hence inherited the basic DOS formats of COM for programs and BMP for images. This is a description of a new Operating System. Some completely new ideas are presented in the OS that needed to be described. Clone OS has a different approach to the common tasks like file management and shell interface. These new ideas are presented here.
Clone OS is a real-mode OS that commonly works in 16-bit with occasional 32-bit usage. It can be extended to long-mode or protected mode.

How Clone OS is loaded –



```
 Clone OS (version 4.7)
View/Editors: load,save,run,text,code,doc,point,icon,paint,video
SettingsCmds: loc,loc2,loc3,prompt,color,color2,drive,drive2,scrolllen,frame
Setting2Cmds: length,difficulty,head,track,page,size
Show Setting: driveinfo,debug,memsize,setting
SetngSwitchs: score,rollcolor,micro,videomode,scrollmode,slowmode,typemode
THE Re-Doers: restart,reboot,restack,reset,cls
OldeAssembly: cli,jmp,hlt,fhlt
Common Keys : Arrow,wasd,F1 series,Tab- Change,E,(Esc or ~) - Close
NewFreshCmds: pipe,install
Experimental: vedit,step,sound,star,print
FATFileComms: z-FDir,x-FSave,fname,a-cmpload,n-Fnew,q-quickload
DirectoryCmd: dir,setdir,newdir
InterruptAPI: 01=step 05=prnscr 20=quit 21,61=api 2c=msg 4a=alarm 60=debug
 Interrupt2 : 1c=timer 03=debug 06=invalid 07=math 1b=ctrl+brk
Time setting: settime,setdate,alarm,alarmtext,autostart
Misc / Extra: htod,calc,clock,play
Andthe Mints: d-date,t-time,c-clock,h-help,v-ver
Moreof Mints: m-mouse,s-space,p-pause,u-up,b-below,o-oh,l-line,e-exit
Bigger Picture:_
```

The first sector of the drive is the bootloader with the last two bytes set as the magic numbers 0x55AA. This tells the BIOS to load it at **0x7C00** location and run it. This bootloader parses the **FAT12** root directory located at **LBA**

0x0013 and finds the OS or asks you to choose one. This entry in root directory tells the first cluster of the file and this can be read until the cluster with value > **0xFF0** is reached. This OS is then run with drive number given in dl register.

Initialization –

Now since the stack could be set anywhere, they need to be initialized.
Clone OS sets stack segment **ss** to **0x9000** and stack pointer **sp** to **0xFFFF**, so we have a large space for stack usage.
After this **ds** is set to zero as well as **es, fs, gs** are initialized to zero.
Now we can save the **dl** value given by the bootloader to the **drive** (system variable).
After the save, we switch to protected mode and back to real mode to get in unreal mode to have both larger memory and BIOS interrupts. All main memory is thus assessable.
In unreal mode, now we set up the IVT with the custom Interrupt handlers for the OS API in **int 0x21,0x61,0x64**, etc and for exceptions **int 0x01,0x03,0x05,0x06**, etc.
After interrupts are set up, a custom user selected autorun string is run.
The autorun commonly contains commands to run -
CONFG (configuration and settings program)
PWD(Password Utility) and
vlh (mints to show version and help).

Shell –

Clone OS has a command line interface. Its shell is tightly combined and mixed in kernel.
Shell starts by showing prompt and receiving a string that can end with enter or space.
These commands are then compared with the list of available commands and if not found, a small error is shown. Shell can be used by programs to run preexisting commands. Clone OS has a flexible approach to all user interface. Thus all Shell colors are customizable. The Default prompt can be changed and custom fonts can be setup to allow all personalization. Shell can be used by programs in the API to use shell commands.

Why Clone OS was built –

It was made and chosen to be a small, portable OS that can run anywhere and used to do everyday jobs with a controlled flow of flexibility and user control over the complete OS. Clone OS was made to be extremely small and run completely from RAM i.e. it’s a completely live OS. It can be run from a -
Hard disk (HDD)
Pen drive (USB)
Memory card (SD)
Floppy disk (FDD)
or any other method that emulates the HDD CHS with 512b sectors.
Books can be read on the move, on any computer using this OS. Also ASCII art was never truly supported by other OS before.
Clone OS has inbuilt ASCII editors with complete saving and creating frames with each ASCII page.
Clone OS also has binary code environment to allow machine coders to code there programs directly with them being saved as a file. This can be used instead of the costly machine kits or simulators.
Clone OS also has DOS compatibility, i.e. COM programs of DOS can be run in Clone OS and vice-versa. This extends to all OS that have a DOS API, including Windows till Xp, React OS and CP/M OS. Clone OS allows more and more support for existing materials and so all existing ebook formats like TXT, DOC are supported and others can be converted to these formats.

Mints –

One of the unique new abilities of Clone OS is Mints.
Mints are characters that can be combined in any manner to make custom commands. Each character in the mint collection is a command. There is no restriction to the length of commands that can be made.
Common mints are - t - time, d-date, p-pause,

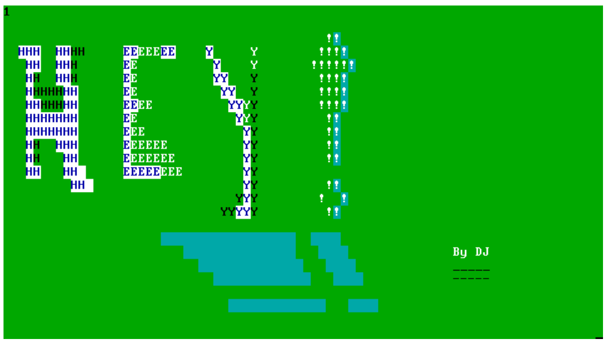
etc.
Thus if we write -
tptdclvllph
it will show time, pause, time, clock, line, show version, line, pause, and finally show help.
The OS has the API setup to allow programmers to use the mints in their own programs.

Directories, programs and files –

All above are treated as file locations. They can be assessed by their file names or their LBA address or CHS sectors. Also each directory is given a unique hexadecimal code, so the common tree structure of assessing files can be skipped and can be jumped to the needed folder immediately.

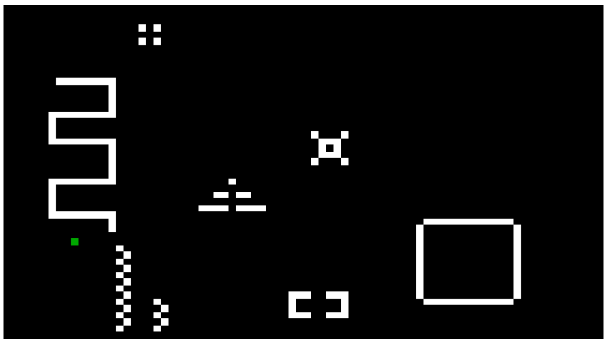
ASCII art –

All 256 characters and 16x16 colors can be used to create any possible painting. Large support exists online for ASCII art by notepad but that allowed only one color and one font.
This ASCII format allows large graffiti to made in small size. E-books can be made with this structure. Also Frames can be combined to make videos. Cards API is provided to allow ASCII art for games.



Applications –

Most common programs are made in COM format. Clock, password system, saved settings, games like ping-pong and snake. Educational simulators like that of stack. And utilities like Screen-savers and portable text and code editors. All Dwarf OS applications can be run directly on Clone OS too. Basic applications like Hex editor, EBook Reader and ASCII editors are inbuilt in this OS. More applications can be simply copied to the drive to add them to the OS.



Microkernel –

Clone OS is also a monolithic kernel with a microkernel. Thus common programs are included in the OS but can be extended to more programs by simply copying the program in the drive. Common utilities are added as files.

Common Keys -
Esc, ~ to Close
F1 for help
Tab to change video page
F10 to quit

Also Ctrl+brk is set up to interrupt any running code to show debug values and pause.
During this pause press F10 to return to shell. This method can be used to close a program forcibly.
This method works like (Ctrl+Alt+Del).

System data -
CHS method - head, track
LBA method - H T S cluster

size - number of sectors to load(used by both CHS and LBA)
color1 - this color is used by the custom text type mode.
color2 -this is used by the BIOS teletype method.

drive - to be used by common loading and saving commands as the drive number
(0x00 = floppy, 0x80 = HDD)
drive2 -install command uses this to copy files from drive1 to drive2.
prompt - default command string.

API –

All of the common tasks faced by the programmer are provided in the API as int 0x21, 0x61, 0x64.

Common API tasks are provided like -

Strings -
display string, get string, read char, show char
Keys-
getch, getch without echo, direct key input

File Management -
open file, open directory, new file, save file, read LBA, and a file selector is provided.

Process Management -
other processes can be run, and closed by different ways. User commands can be piped to allow instruction processing with different modules.
Debugging options are provided to you in int 0x60 or opcode 0xCC can be used for a breakpoint.

TUI components like -
Message Box, Input Box, save screen and restore are provided.

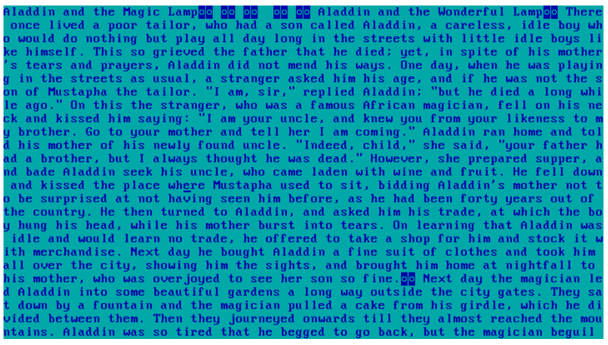
And all other everyday components required by programmers like -

clrscr, newline, position setup, and getting integers or bigger numbers is already set to be quickly used by the developers.

Problems faced -

No good e-book format was found to provide a good reading experience in smaller sizes.
Pictures have to be set in books but this made large sized books and lesser creative control.
This was solved by creating a new book type .PIC which can contain both text and pictures in very small space. It also allows variable experimentation features to be used.

Also too less .COM programs had existed, so our developers sat straight to remake the basic utilities.



```
Aladdin and the Magic Lamp◙◙ ◙◙ ◙◙  ◙◙ ◙◙ Aladdin and the Wonderful Lamp◙◙ There
 once lived a poor tailor, who had a son called Aladdin, a careless, idle boy wh
o would do nothing but play all day long in the streets with little idle boys li
ke himself. This so grieved the father that he died; yet, in spite of his mother
's tears and prayers, Aladdin did not mend his ways. One day, when he was playin
g in the streets as usual, a stranger asked him his age, and if he was not the s
on of Mustapha the tailor. "I am, sir," replied Aladdin; "but he died a long whi
le ago." On this the stranger, who was a famous African magician, fell on his ne
ck and kissed him saying: "I am your uncle, and knew you from your likeness to m
y brother. Go to your mother and tell her I am coming." Aladdin ran home and tol
d his mother of his newly found uncle. "Indeed, child," she said, "your father h
ad a brother, but I always thought he was dead." However, she prepared supper, a
nd bade Aladdin seek his uncle, who came laden with wine and fruit. He fell down
 and kissed the place where Mustapha used to sit, bidding Aladdin's mother not t
o be surprised at not having seen him before, as he had been forty years out of
the country. He then turned to Aladdin, and asked him his trade, at which the bo
y hung his head, while his mother burst into tears. On learning that Aladdin was
 idle and would learn no trade, he offered to take a shop for him and stock it w
ith merchandise. Next day he bought Aladdin a fine suit of clothes and took him
all over the city, showing him the sights, and brought him home at nightfall to
his mother, who was overjoyed to see her son so fine.◙◙ Next day the magician le
d Aladdin into some beautiful gardens a long way outside the city gates. They sa
t down by a fountain and the magician pulled a cake from his girdle, which he di
vided between them. Then they journeyed onwards till they almost reached the mou
ntains. Aladdin was so tired that he begged to go back, but the magician beguil
```

Future Work -

FAT16 is planned to be added as another driver.
Mouse support can be added to int 0x33 .
GUI mode can be made additional to current CLI.
EXE files can be run.

Additional compilers for higher level languages.
Other architectures like SPARC and ARM can be supported.

References -

[1]Silberschatz, Abraham, et al. *Operating system concepts*. Vol. 4. Reading: Addison-Wesley, 1998.
[2]Hoare, Charles Antony Richard. "Monitors: An operating system structuring concept." *Communications of the ACM* 17.10 (1974): 549-557.
[3]Tanenbaum, Andrew S. *Modern operating systems*. Prentice Hall Press, 2007.
[4]McKusick, Marshall Kirk, et al. *The design and implementation of the 4.4 BSD operating system*. Pearson Education, 1996.
[5]Leslie, Ian M., et al. "The design and implementation of an operating system to support distributed multimedia applications." *Selected Areas in Communications, IEEE Journal on* 14.7 (1996):

1280-1297.
[6] Linux scalability e ort. http://lse.sourceforge.net/.
[66] A. S. Tanenbaum and R. van Renesse. Distributed operating systems.
ACM Computing Surveys, 17(4):419–470, 1985.
[7]. Broken Thorn Operating System Development Series
[8]. OS Dev.org