Whanted to get a openvms dev enviroment, where could use it to learn the basic command's to get feel for it without worrying about breaking anything.

1. So whanted to know if its possible to install a openvms iso image in Virtual Box
2. In case its possible where could get a openvms image or needed first to get installation cds and from those generate the iso?
3. if the cds are needed is there some online store that can be bought or the price is only afordable for companies?

only making inqueries to see if its possible and feasible, other wise gona give up the ideia of having a development enviroment.

------------------------------------------------Answer--------------------------------------------------------------------

Well what do you know so far, and what do you have so far? Do you know that OpenVMS has a 40+ year history, was originally written for the (32 bit) VAX platform, ported to 64-bit Alpha (30+ years ago), then ported to Itanium and is now (finally?!) being ported to run native on X86 (64 bit). That X86 port will boot natively and you may want to ue VMware, Virtualbox or similar - within the current constraints and offeringg - visit vmssoftware.com for details.

For initial learning I recommend to just to use an Alpha or VAX emulator. Those could run on a virtual machine, but there is no good reason. Just run as process on whatever laptop/server your have running Windows (and some Linux options I think).

There are are several emulators out there, both free and commercial - google is your friend. I happen to like/use FreeAXP for Alpha running under 64 bit windows - just Google.

There is no Itanium emulator.

Do you have access to the software distribution somehow?

For starters, why even bother trying to get your own system? Just timeshare to get a feel for it. For example SSH to decuserve.org [184.168.131.241] - ask for an account - presto! Other timeshare options are available - and very valid to learn the basics (file system, editors, compilers,...)

Good luck! Hein.

---------------------------------------Main Commands-------------------------------------------------

Digital Command Language (DCL) --- DCL commands (sometimes called *verbs*)

Commands are not case sensitive

There is no concept in OpenVMS like the UNIX hash table

VERB DIRECTORY

help hints

DIRECTORY, PRINT, and SEARCH) and can be abbreviated, in these cases to DIR, PRI, and SEA

SHOW TIME

Command qualifiers (sometimes called *switches*) modify the action of the verb

$ HELP /MESSAGE (to provide a more detail information about error of a command)

Example of the command and error:

$ Man show %DCL-IVVERB, unrecognized command verb - check validity and spelling \MAN\

Then the command to get more information would be

$ help/mess ivverb

It will print pages 3 through 6 of the specified file and display the file's name on the top of each page.

$ PRINT /HEADER/PAGE=(3:6) MYFILE.TXT

A fully qualified ODS-2 file name has the following format:

node::device:[directory.sub-dir.sub-sub-dir]file.extension;version

Files formats and extensions:

* COM is a command procedure file.
* CC is a C source file.
* DIR is a directory file. The UNIX convention,. and .., is not used in OpenVMS.
* EXE is an executable file created by LINK.
* OBJ is an object file created by a compiler.

 DCL commands such as COPY also create a new version

File characteristics are changed with the SET FILE command. Thus, the UNIX commands—chmod, chown, and chgrp—are all rolled into SET FILE. Because OpenVMS supports so many different file types, SET FILE has many more capabilities.

Symbols are created with the assignment operation. Most often they are defined in command procedures.

 It creates a new symbol, LS (which, after it is defined, can be used like a verb), to customize the DIRECTORY command:

$ LS :== DIR/SIZE/COL=1

Incidentally, in OpenVMS, file sizes are measured in units of blocks, not characters. A *block* is 512 characters.

Logicals and symbols are often confused. In simplistic terms, there are two differences:

1. *Logicals are recursive, but symbols are not*. That is, a logical may be defined using another logical, but a symbol defined in this way cannot contain another symbol.
2. *Logicals are translated anywhere in the command line, but symbols are translated only at the left side of the command*. This means DIR SYSUAF and LS are the proper usage for the definitions. DIR LS would result in an error message.

DIRECTORY -> dir (List of all files)

Dir \*.dir (List of directories that is the filter applied \* stands for ant and then the extension dir)

SET DEFAULT [HOMEDIR.SUBDIR2.SUBDIRC] (change to another directory)

$sh log ‘logicalname’ -> SHOW LOGICAL ‘logicalname’

$sh sym ‘symbol name’ -> SHOW SYMBOL ‘symbol name’

$dir \*,dir #returns all directories in current directorie

$dir /since #filters and shows all file created or edited today its also possible to specify a date as /since=YYYYMMDD mas confirmer o formato

$ dir ‘filename’ /FULL #returns the properties of the file in more detail