SEARCH ORDER

Internal vs External Commands

As described in an earlier lecture, commands can be broken into two main categories: internal and external.

An internal command is one that is built-in to the command interpreter (cmd.exe); when cmd.exe is launched all its built-in commands are loaded into and reside in RAM. They are always stored in memory and therefore easily accessible by the operating system. When executing an internal command, the operating system quickly finds the command in memory and executes it.

An external command is one that is contained within an executable file on disk.
When executing an external command, the operating system must find the file on disk and load it into memory before it can be executed.

> An executable file is a file that has an extension of .com, .exe, .bat, .cmd. This means that the operating system would actually be searching for any one of the following files: *commandused*.com; *commandused*.exe, *commandused*.bat or *commandused*.cmd. There are additional file types that will be searched for but typically an executable is identified as a file with an extension of .exe, .com, .bat or .cmd The additional file extensions that will be searched for will vary by computer and can be viewed by typing echo %pathext%.

Search Order

When executing any command from the command prompt the operating system must find the command before it can be executed. Here's how the search works:

1) It searches Memory (RAM) – if the command is an internal command it will be found in memory and be executed. The search would stop here.

2) If the command is not found in memory, then the operating system will look for an executable file by that name in the directory from which you're executing the command; your current location. If one is found, it is loading into memory and executed and the search would stop here.
   Example: if your current location is C:\DATA, the operating system will search the C:\DATA directory for an executable file that matches the name of the command that you typed.).

3) If an executable file by that name is not found in the current location, the operating system will begin sequentially searching each directory listed in the *search path*.
   The PATH environment variable defines the Windows search path. The search path is an ordered, colon-separated list of directories that are searched, left to right, for the executable command that you typed.

   View the value of your search path by typing PATH.

   Example:
   C:\>path
   PATH=c:\windows\system32;c:\mystuff

   The operating system will search the first directory in the path (in this case, the *c:\windows\system32* directory); if the file is found there, it is loaded from the

hard drive into RAM and executed. If it is not found, the operating system will search the next directory in the path (in this case, the *c:\mystuff* directory). If the file is found there, it's loaded into memory and executed. In this example there's only two directories; if there were more directories, the search would continue through each of the directories in the list.

If the file is not found in any of the directories in the Search Path, the user would receive a message indicating "*command is not recognized as an internal or external command operable program or batch file).*

The examples of path values presented in this document are not typical of what you'd find on most modern Windows computers. The objective of these examples is to help you understand how the search order works. A modern Windows system would have a much longer value in its search path and would always include the C:\windows\system32 directory.

Example 1:

PATH=C:\WINDOWS\SYSTEM32;C:\NOVELL\CLIENT32;C:\PROGS

Current location: G:\MYFILES>

Assume the following command is executed

G:\MYFILES>CONFIG Note: this is not a valid command

Here's what would happen:

1) The operating system would check memory for this command. It's not an internal command so the search would continue.
2) The default directory (g*:\myfiles>)* would be searched for an executable file with the name of CONFIG (i.e., config.exe, config.com, config.bat, config.cmd..). The file would not be found in this directory therefore the search would continue.
3) The *C:\windows\system32* directory would be searched for an executable file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
4) The *C:\novell\client32* directory would be searched for an executable file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
5) The *C:\progs* directory would be searched for a executable file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
6) Since there are no directories remaining in the search path and the file was not found, the message '*the command is not recognized as an internal or external command operable program or batch file"* would be displayed

Example 2:

PATH=C:\WINDOWS\SYSTEM32;C:\NOVELL\CLIENT32;C:\PROGS

Assume the following command is executed

G:\MYFILES>attrib Note: this is a valid external command

Here's what would happen:

1) The operating system would check memory for this command. It's not an internal command so the search would continue.
2) The current location (g*:\myfiles>)* would be searched for an executable file with the name ATTRIB. The file would not be found in this directory therefore the search would continue.
3) The *C:\windows\system32* directory would be searched for an executable file with the name ATTRIB. The file *would* be found in this directory therefore the operating system would load the file from disk into memory and execute the command.

Example 3:

PATH=C:\;C:\NOVELL\CLIENT32;C:\PROGS

Current Location: C:\WINDOWS\SYSTEM32>

Assume the following command is executed

C:\WINDOWS\SYSTEM32>CHKDSK Note: this is a valid external command

Here's what would happen:

1) The operating system would check memory for this command. It's not an internal command so the search would continue.
2) The current location (*C:\windows\system32>)* would be searched for an executable file with the name CHKDSK. The file *would* be found in this directory therefore the operating system would load the file from disk into memory and execute the command.

Example 4:

PATH=C:\;C:\NOVELL\CLIENT32

Current Location: C:\MYFILES>

Assume the following command is executed

C:\MYFILES>CHKDSK Note: this is a valid external command

Here's what would happen:

1) The operating system would check memory for this command. It's not an internal command so the search would continue.
2) The current location (*C:\myfiles>)* would be searched for an executable file with the name of CHKDSK. The file would not be found in this directory therefore the search would continue.
3) The root of drive C: (*C:\)* would be searched for an executable file with the name of CHKDSK. The file would not be found in this directory therefore the search would continue.
4) The *C:\novell\client32* directory would be searched for an executable file with the name of CHKDSK. The file would not be found in this directory therefore the search would continue.
5) Since there are no directories remaining in the search path and the file was not found, the message '*the command is not recognized as an internal or external command operable program or batch file"* would be displayed indicating that the command would not found

Example 5:

PATH=C:\;C:\NOVELL\CLIENT32

Current location: C:\MYFILES>

Assume the following command is executed

C:\MYFILES>DIR

Here's what would happen:

1) The operating system would check memory for this command. It *is* an internal command therefore the operating system would execute the command from RAM.

PATH COMMAND

The below table provides examples of using the PATH command. For the syntax of the command, type PATH /? at the command prompt or HELP PATH

When using the PATH command to change the value of the search path, it is temporary to the current command prompt session. If you close the window and reopen another command prompt window, the system's default value for the PATH will automatically be re-established.

PATH	Displays the existing search path setting
PATH C:\windows\system32;c:\windows;c:\progs	Sets the PATH to search the three directories listed (C\windows\system32, c:\windows and C:\progs). A semi colon (;) is used to separate directory names in a path.
PATH %PATH%;C:\progs	The %path% variable stores the current path. Using this variable in the PATH command provides a means of adding an additional directory to an existing path. In this example, the C:\progs directory would be added to the ***end*** of the existing path's value.
PATH C:\PROGS;%PATH%	The %path% variable stores the current path. Using this variable in the PATH command provides a means of adding an additional directory to an existing path. In this example, the C:\progs directory would be added to the *beginning* of the existing path's value.
PATH ;	Clears the existing search path setting. This would direct the command interpreter to search only the current directory (after searching RAM) when searching for executables.

As noted above, when using the PATH command, as presented in the above table, it will temporarily change the value of the search path within that command prompt session. The value can be permanently changed with the SETX command (not covered in this course) or through system properties: sysdm.cpl / Advanced / Environment Variables. The search path is used by the command interpreter and it's also used by many installed programs. Point being, don't make permanent changes to this value if you are not sure what you're doing.