PRUEBA ABIERTA: UD3

GESTIÓN DE LA INFORMACIÓN

Francisco Javier Martínez Reguera

ÍNDICE

1. INTRODUCCIÓN

En este documento se tratarán varios temas relacionados con la gestión de la información, como son las variables de entorno y la seguridad informática. Veremos para qué se utilizan las variables de entorno en los sistemas operativos, las variables de entorno más conocidas de Windows y analizaremos por qué se deben proteger los datos de las empresas tanto física como lógicamente, indicando varias maneras de hacerlo.

2. VARIABLES DE ENTORNO: USOS Y EJEMPLOS

Las variables de entorno son cadenas de texto que sistemas operativos como Windows, Linux o Mac usan para almacenar valores que normalmente hacen referencia a archivos, directorios y funciones comunes del sistema cuya ruta concreta muchos programas necesitan poder conocer. Casi cualquier usuario medio de ordenadores ha tenido que lidiar en alguna ocasión con variables de entorno. Por ejemplo, puede ser que a la hora de instalar o configurar algún programa para Windows nos hayamos encontrado con una ruta que, en lugar de seguir la típica estructura "C:/Users/Javier/Documents", mostrase algo como "%USERPROFILE%/Documents". El comando %USERPROFILE%, el cual es una variable de entorno, permite que un programa sepa acceder a nuestra carpeta de usuario incluso si no se le ha indicado el nombre del mismo o si no sabe qué versión de Windows estamos usando. Para conocer el valor de algunas de estas variables podemos ingresarlas directamente en la consola de CMD, en Inicio o en Ejecutar, mientras que para mostrar otras tenemos que anteponer el comando "echo" a la variable.

Se debe recalcar que las variables de entorno no siempre equivalen a rutas de directorios, sino que pueden remitir a otra clase de información. Así, %TIME% devuelve la hora actual del sistema, %OS% la versión del sistema operativo y %PATHEXT% la lista de extensiones de archivo consideradas ejecutables. A continuación se muestra una lista con las variables de entorno más empleadas en Windows. Se debe tener en cuenta que puede haber ligeras variaciones en el valor devuelto según la versión del sistema operativo.

Variable de entorno	Descripción
%ALLUSERSPROFILE%	Devuelve la localización del perfil de todos los usuarios, por lo general C:\Program\Data, carpeta oculta en la unidad C.
%APPDATA%	Devuelve la localización de una carpeta oculta donde guardan sus datos las aplicaciones, dentro de la carpeta de usuario. En Windows 7 y Vista la ruta es la siguiente: C:\Users\NombreDeUsuario\AppData\Roaming. En Windows XP la ruta es diferente: C:\Documents and Settings\NombreDeUsuario\Datos de programa.

%COMMONPROGRAMFILES%	Devuelve la localización de una carpeta donde los programas almacenan archivos comunes: C:\Program Files\Common Files.
%CMDCMDLINE%	Muestra el comando exacto empleado para acceder al intérprete de comandos.
%CMDEXTVERSION%	Devuelve el número de versión de nuestro intérprete de comandos.
%COMPUTERNAME%	Devuelve el nombre del equipo.
%COMSPEC%	Devuelve la ruta de la *shell* de comandos, normalmente C:\Windows\System32.
%DATE%	Devuelve la fecha actual.
%ERRORLEVEL%	Devuelve el código de error del último comando ejecutado.
%HOMEDRIVE%	Devuelve la unidad en la que está el directorio en el que estás actualmente.
%HOMEPATH%	Devuelve la ruta completa a dicho directorio.
%LOGONSERVER%	Devuelve el nombre de nuestro servidor.
%LOCALAPPDATA%	Carpeta donde los programas guardan archivos temporales, generalmente C:\Users\NombreDeUsuario\AppData\Local.
%NUMBER_OF_PROCESSORS%	Devuelve el número de procesadores instalados en el equipo.
%OS%	Devuelve nuestro sistema operativo.
%PATH%	Contiene una lista separada por punto y coma de directorios en los cuales se buscarán los archivos ejecutables que no se invocan con una ruta explícita. Aparecerán los siguientes directorios más otros agregados por diversas aplicaciones: C:\Windows\system32, C:\Windows, C:\Windows\System32\Wbem, C:\Windows\System32\WindowsPowerShell\v1.0\.
%PATHEXT%	Contiene una lista separada por punto y coma con las extensiones conocidas de los archivos ejecutables. Si el nombre de un ejecutable termina con una extensión incluida en esta lista, es posible omitir dicha extensión al invocar el programa.
%PROCESSOR_ARCHITECTURE%	Devuelve la arquitectura del procesador.
%PROCESSOR_IDENTIFIER%	Devuelve la descripción del procesador.
%PROCESSOR_LEVEL%	Devuelve el número de modelo de procesador.
%PROCESSOR_REVISION%	Devuelve el número de revisión del procesador.
%PROGRAMDATA%	Devuelve la localización de una carpeta donde los programas almacenan datos, normalmente C:\ProgramData.

%PROGRAMFILES%	Devuelve la carpeta donde se instalan los programas, normalmente C:\Archivos de Programa.
%PSModulePath%	Ruta a los módulos de PowerShell, C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
%PUBLIC%	Carpeta donde se guardan datos públicos compartidos de todos los usuarios, generalmente C:\Users\Public.
%RANDOM%	Devuelve un número al azar entre 0 y 32767.
%SYSTEMDRIVE%	Devuelve la unidad que contiene el directorio raíz del sistema, generalmente C.
%SYSTEMROOT%	Devuelve la carpeta de administración, que suele ser C:\Windows.
%TEMP% y %TMP%	Contienen la ruta a los directorios donde las aplicaciones y programas pueden almacenar sus archivos temporales. En Windows 7 es: C:\Users\NombreDeUsuario\AppData\Local\Temp. En Windows XP la ubicación es: C:\Documents and Settings\NombreDeUsuario\Configuración local\Temp.
%TIME%	Devuelve la hora actual.
%USERNAME%	Devuelve el nombre del usuario actual.
%USERPROFILE%	Devuelve la ruta del directorio donde están los archivos del usuario actual, normalmente C:\Users\NombreDeUsuario.
%WINDIR%	Devuelve la ruta de la carpeta del sistema operativo, normalmente C:\Windows.

La variable de entorno con la que más habitualmente tendremos que lidiar será %PATH%. Cuando tecleamos un comando propio de Windows no es necesario teclear la ruta completa del ejecutable ya que, cada vez que tecleamos un comando, el sistema revisa las carpetas contenidas en la variable %PATH% para comprobar si algún archivo ejecutable coincide con el mismo, de ahí su importancia. Es un recurso muy usado, por ejemplo, para los desarrolladores que desean llamar a un intérprete o compilador desde la carpeta del proyecto en el que están trabajando. Muchos IDE también recurren al %PATH% para ejecutar dichas herramientas.

3. SEGURIDAD INFORMÁTICA

Podemos entender como seguridad a una característica de cualquier sistema (informático o no) que nos indica que ese sistema está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible. La pérdida tanto de material electrónico como de información interna es una situación nada deseable tanto para un usuario individual como para una empresa, de ahí su importancia.

La seguridad física de un sistema informático consiste en la aplicación de barreras físicas y procedimientos de control frente a amenazas físicas al hardware. Este tipo de seguridad está enfocada a cubrir las amenazas ocasionadas tanto por el hombre como por la naturaleza del medio físico en que se encuentra ubicado el sistema. Ejemplos típicos de amenazas de tipo físico son, por ejemplo, terremotos, incendios accidentales, inundaciones, robos, sabotajes, señales electromagnéticas, apagones o sobrecargas. Tener controlado el ambiente y acceso físico permite disminuir siniestros, y tener los medios para luchar contra ellos reduce los daños una vez ocasionados. Cada tipo de amenaza tendrá sus propias medidas de prevención. Para evitar incendios se puede recurrir a mobiliario ignífugo, detectores de humo, extintores, etc. En cuanto a las inundaciones, se recomienda, por ejemplo, evitar ubicar los equipos electrónicos en plantas bajas. Los robos pueden reducirse con una buena vigilancia, ya sea presencial o mediante cámaras. Para evitar daños por ondas electromagnéticas se deben evitar las zonas con grandes emisiones de radiación o utilizar equipos y materiales protectores contra las interferencias en caso de que no sea posible. Para evitar apagones se puede recurrir a sistemas de alimentación ininterrumpida.

La seguridad lógica de un sistema informático consiste en la aplicación de barreras y procedimientos que protejan el acceso a los datos y a la información contenida en él. Los robos de datos pueden prevenirse cifrando la información almacenada y usando contraseñas o sistemas biométricos para acceder a ella. La pérdida de información también es una amenaza lógica frecuente. Para evitarla, se puede recurrir a copias de seguridad. Otra amenaza frecuente son los virus, a los que se puede plantar cara mediante el uso de antivirus. Los firewall y servidores proxy serán esenciales para evitar ataques desde la red, autorizando y auditando las conexiones permitidas.

4. CONCLUSIONES

Se ha visto que las variables de entorno son cadenas de texto que almacenan valores que no solo hacen referencia a archivos y directorios, sino también a funciones y características del sistema, dándose algunos ejemplos pertenecientes a Windows. Por otra parte, se ha definido seguridad informática así como sus ramas física y lógica, dándose varios ejemplos de amenazas asociadas a cada una de ellas y la forma de prevenirlas.

BIBLIOGRAFÍA

[1] *Sistemas Informáticos*, Apuntes de Cesur.

[2] https://es.wikipedia.org/wiki/Seguridad_inform%C3%A1tica (Última visita: 30/11/2020)

[3] https://es.wikipedia.org/wiki/Variable_de_entorno (Última visita: 30/11/2020)

[4] https://www.genbeta.com/desarrollo/variables-entorno-que-sirven-como-podemos-editarlas-windows-linux (Última visita: 30/11/2020)

[5] https://norfipc.com/inf/variables-entorno.html (Última visita: 30/11/2020)

[6] http://e-ducativa.catedu.es/44700165/aula/archivos/repositorio/1000/1063/html/index.html (Última visita: 30/11/2020)

[7] https://www.fpgenred.es/Seguridad-Informatica-I/seguridad_fsica_y_lgica.html (Última visita: 30/11/2020)