

Por qué un lenguaje de programación sólo puede utilizarse en algunos sistemas operativos y en otros no

Porque un lenguaje de programación compila a lenguaje ensamblador el cual es ejecutado por el sistema operativo, este lenguaje ensamblador depende de 2 cosas, de la arquitectura del procesador y la interpretación que le de el sistema operativo. Si en la computadora cambia la arquitectura del procesador el lenguaje ensamblador es totalmente diferente y no va a correr.

Ahora sí es la misma arquitectura de procesador y diferente sistema operativo, en ocasiones el sistema operativo cambia la definición de la interrupción o el llamado de esta; por ejemplo en Windows para mostrar en pantalla usas la interrupción 21 (si no mal recuerdo), y en Linux me parece que es la 10 (si recuerdo bien), lo cual hace los ejecutables finales incompatibles entre sistemas operativos.

A2A. Hola, bueno, la verdad yo no conozco lenguajes de programación con esas características (no digo que no existan, solo que yo no los conozco), lo más parecido a lo que tu dices, a mi parecer, sería la sintaxis de la línea de comandos de cada sistema operativo —la cual al ser diseñada para una familia de sistemas operativos específicos, lógicamente no funcionará para otros (al menos no de forma nativa)— o en tal caso quizás un lenguaje de scripts —cuyo interprete solo esté disponible para algunos sistemas operativos y para otros no, lo cual no permite que se puedan ejecutar dichos scripts en tales sistemas—, por ejemplo. Cabe destacar que hay, lenguajes de programación como Ensamblador, el cual es dependiente de la arquitectura del procesador, por lo que un programa escrito en este lenguaje, difícilmente se podrá compilar en otro tipo de procesador y sistema operativo diferentes para los que fué escrito; no se si a esto es a lo que te referías, pero en fin espero te sirva la respuesta, saludos.

Cualquier lenguaje de programación puede usarse en un sistema siempre y cuando haya un intérprete o compilador (según corresponda al lenguaje en cuestión) si es que deseas programar en esa máquina con ese sistema operativo. Obvio si no hay tal intérprete o compilador no podrás usar ESE, tu lenguaje favorito, en ese sistema.

Además, por otro lado, es factible utilizar una técnica llamada cross-compile en donde en una máquina y sistema operativo "X" puedes compilar código objeto (léase, binarios ejecutables) para otra máquina o sistema diferente "Y". Al decir "diferente" puede ser: diferente sistema operativo, diferente procesador y, en casos extremos hasta diferentes arquitecturas de procesador.

Después de todo, las nuevas tecnologías son diseñadas por otros sistemas, precursores, mas viejos si lo quieres ver así.

¿Qué tipo de máquina virtual en VirtualBox?

**VirtualBox** es una aplicación que sirve para hacer máquinas **virtuales** con instalaciones de sistemas operativos. Esto quiere decir que si tienes un ordenador con Windows, GNU/Linux o incluso macOS, puedes crear una **máquina virtual** con cualquier otro sistema operativo para utilizarlo dentro del que estés usando.

En otras palabras, esto te va a permitir instalar otros sistemas operativos o el mismo que ya tienes dentro de tu ordenador. El primer caso te va a servir para probar las aplicaciones de otro sistema operativo para no tener que estar haciendo particiones o el proceso de instalar Linux junto a Windows 10. No es una experiencia tan nativa y buena como tener instalado el sistema operativo en el ordenador, pero te va a servir para hacer pruebas puntuales.

Si tengo más de una máquina virtual instalada, y una se rompe, ¿esto afecta a las demás? ¿por qué?

Las máquinas virtuales de procesos se ejecutan en un mismo servidor para ejecutar varias instancias de la misma aplicación de forma separada. De ese modo, **si una de ellas falla, no afectará al funcionamiento del resto**

<https://kb.vmware.com/s/article/1007819?lang=es>

```
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 8.124/8.941/9.759/0.822 ms
usuario@ubuntu-Intro:~$ sudo passwd root
[sudo] password for usuario:
Lo sentimos, vuelva a intentarlo.
[sudo] password for usuario:
Lo sentimos, vuelva a intentarlo.
[sudo] password for usuario:
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
usuario@ubuntu-Intro:~$ apt-get install git
E: No se pudo abrir el fichero de bloqueo «/var/lib/dpkg/lock-frontend» - open (13: Permiso denegado)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
usuario@ubuntu-Intro:~$ sudo apt-get install git
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch
  git-cvs git-mediawiki git-svn
Se actualizarán los siguientes paquetes:
  git
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 183 no actualizados.
Se necesita descargar 3.312 kB de archivos.
Se utilizarán 32,8 kB de espacio de disco adicional después de esta operación.
Des:1 http://co.archive.ubuntu.com/ubuntu xenial-updates/main i386 git i386 1:2.7.4-0ubuntu1.10 [3.3
12 kB]
Descargados 3.312 kB en 1s (1.804 kB/s)
(Leyendo la base de datos ... 58124 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../git_1:2.7.4-0ubuntu1.10_i386.deb ...
Desempaquetando git (1:2.7.4-0ubuntu1.10) sobre (1:2.7.4-0ubuntu1.6) ...
Configurando git (1:2.7.4-0ubuntu1.10) ...
usuario@ubuntu-Intro:~$ git --version
git version 2.7.4
usuario@ubuntu-Intro:~$ _
```



