



Que el alumno vea que no sólo existe la programación clásica de software, sino que tiene la posibilidad de controlar y programar tareas que ejecute su propio ordenador de manera automática de manera que el propio ordenador se encargue del trabajo aburrido y el alumno se pueda dedicar a tareas más atractivas sin perder tiempo en trabajos repetitivos con poco valor añadido

Convertir una computadora tonta en algo más inteligente: bash scripting

La fusión perfecta entre programación y el intérprete de comandos más común de Linux es el *bash scripting*. Mediante el uso de bash podremos realizar tareas con la computadora que mediante la *programación clásica* no es posible o es más complejo. Bash es una herramienta popular de scripts disponible en Unix y son un componente muy potente y útil para el desarrollo de automatismos. Generalmente usan intérpretes en lugar de compiladores para ejecutarse.

Todxs somos conscientes de que realizar tareas repetitivas tiene varios problemas:

- 1.- Es tedioso.
- 2.- Estás abocado al error.
- 3.- Es una pérdida de tiempo.

Sin embargo, si eres capaz de automatizar una tarea repetitiva, no te deberías equivocar y la computadora debería trabajar por tí; suena útil ¿verdad?

Mediante instrucciones bash el administrador de la computadora:

- 1.- Podrá automatizar acciones repetitivas, gracias al uso de expresiones matemáticas, como condiciones, bucles, etc.
- 2.- El resultado de estos mismos automatismos llevará a que el sistema operativo sea más automático y más ágil.
- 3.- La persona a cargo se podrá dedicar a otras tareas.

Esto es una maravilla! Además crear scripts en bash es una tarea bastante sencilla y poderosa al mismo tiempo. Con un sencillo script se pueden optimizar procesos y automatizarlos para dar solución a problemas muy concretos. En muchas ocasiones el bash scripting está muy asociado a la administración de servidores pero en tu computadora doméstica también puedes construir tus propios automatismos. La única limitación debería ser tu propia imaginación. ¿Qué tarea te gustaría automatizar?

Aunque puedes llegar a pensar que los scripts en bash están pensados para administradores de sistemas, lo cierto es que lo puedes emplear para cualquier cosa que te puedas imaginar:

- 1.- Puedes automatizar la conversión a PDF, cada vez que edites un archivo.
- 2.- Puedes hacer copias de seguridad cada X tiempo y eliminar las copias antiguas.
- 3.- Leer ficheros y tomar decisiones según el resultado
- 4.- Arrancar determinados programas al encender el ordenador.
- 5.- Enviar mensajes automáticos de control en determinados momentos.
- 6.- Rastrear el estado de la red de casa haciendo ping contra todas las ip-s.
- 7.-

...

Estas son sólo algunas razones para ponerte manos a la obra y aprender a crear tus propios scripts en Bash. Para picar *code* en bash deberías ejecutar la shell, que es el interprete de comandos del Sistema Operativo. Suele ser una interfaz de texto de altas prestaciones que se manifiesta en forma de Terminal (Consola) y que sirve esencialmente para 3 importantes áreas de trabajo:

- 1.- Administrar el Sistema Operativo
- 2.- Ejecutar aplicaciones e interactuar con ellas
- 3.- Servir de entorno básico de programación.

El scripting se refiere a la técnica de diseñar y crear scripts mediante un Shell para la generación de automatismos que nos faciliten la vida y son extremadamente útiles. Podría ser una buena idea escribir aquellas necesidades que tengamos en un papel y luego editar scripts que realicen este trabajo por nosotros. ¿Te animas?

Tarea:

- 1.- Escribir en un editor tres ideas que te gustaría automatizar.
- 2.- Abrir el Ubuntu u otra máquina virtual instalada. Cómo instalar una máquina virtual: Tarea 51
- 3.- Hacer los programas.
- 4.- Subir toda la documentación a tu repositorio Github.

#HASHTAGS *(etiquetas de ayuda para búsqueda de información relevante)*

#bash-scripting #shell #linux #unix #consola

LINKS DE INTERÉS

<https://openwebinars.net/blog/que-es-bash-scripting/>
<https://www.hostinger.es/tutoriales/bash-script-linux>
<https://likegeeks.com/es/script-de-bash-tutorial/>
<https://www.atareao.es/tutorial/scripts-en-bash/>
<https://blog.desdelinux.net/shell-scripting-archivo-desktop/>

DICCIONARIO

Bash | Shell | Script | Interprete | Compilador

PUNTUACIÓN

Programación: 4

Redes: 4

Seguridad: 3

Algoritmia: 2