

Escritura de scripts Shell en Unix/Linux

Curso práctico de 3 días - 21h

Ref.: SHL - Precio 2025: 1 450€ sin IVA

El shell es a la vez un lenguaje y un programa, y aquí le presentamos su comportamiento detallado para que pueda dominar su programación. Este curso le permitirá adquirir verdadera autonomía en la escritura de scripts de shell para aplicaciones específicas (supervisión, automatización, instalación de software, tratamiento de archivos, etc.).

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Conocer las diferentes instrucciones que se pueden utilizar en un script de shell

Escribir scripts sencillos de funcionamiento de Unix/Linux

Depuración de un script de shell

Enriquecer un script de shell con funciones y subshells

Manipulación de archivos con los comandos grep, find, sed y awk

Se realizarán varios ejercicios (sencillos y complejos) para ayudar a los alumnos a ser realmente independientes en la escritura de scripts de shell.

PROGRAMA

última actualización: 02/2024

1) Presentación y recordatorios

- Los diferentes intérpretes: Bourne Shell, Korn Shell, Bash, C Shell, Tcsh, etc.
- Disponibilidad de intérpretes en los distintos sistemas Unix.
- Puesta al día sobre la normalización (impacto en los scripts).
- Entradas GNU (gawk, gsed...).
- Diferencias entre Bourne Shell/Korn Shell/Bash.

2) Programación de guiones

- Herramientas de desarrollo.
- Mecanismo de ejecución de scripts.
- Reglas de búsqueda de pedidos.
- Principios de ejecución de comandos (exec, pipeline, subshell, background, etc.).
- Principios de ejecución de guiones.

3) Mecanismos básicos

- Lectura y análisis de la línea de comandos.
- Ampliación de llaves, desarrollo de la tilde, sustitución de parámetros.
- Sustitución de órdenes y evaluación aritmética.
- Procesos de escape (normalización).
- Redirecciones (entrada y salida estándar, archivos, tubos, documentos en línea).

4) Funcionamiento interactivo

- Invocación del shell (opciones).
- Los diferentes archivos de puesta en marcha.
- Conceptos de entorno (variables, alias, funciones).

PARTICIPANTES

Desarrolladores, técnicos y administradores de Unix/Linux.

REQUISITOS PREVIOS

Conocimientos básicos de un sistema Linux/Unix y de programación. Se valorará la experiencia en el uso de uno de estos sistemas.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

- Historial de pedidos y retirada.
- Control del trabajo.
- Terminación del sustantivo.
- Terminación de la concha.

5) Creación de scripts de shell portables (ksh/bash)

- Interfaz shell-script.
 - Estructuración de un shell-script. Nociones de subshell.
 - Llamar a un intérprete de comandos: los diferentes métodos. Opciones disponibles. Más información sobre el entorno. Opciones y argumentos.
 - Preámbulo del shell-script: ¿quién interpreta el shell-script? Comentarios sobre este artículo.
 - Parámetros de posición (inicialización, guardado, desplazamientos).
 - Variables locales. Variables globales.
 - Declaración y visibilidad de las funciones.
 - Salida Shell-script. Función de salida. Convenciones utilizadas. Valor de retorno.
- Encadenamiento de scripts de shell.
- Estructuras de control de Shell: comandos simples, canalizaciones, listas de canalizaciones.
 - Comandos compuestos, subshells y funciones. Mecanismos de selección e iteración.
- Menús.
- Entradas/salidas. Interacción con el sistema. Argumentos de la línea de comandos.
 - Operaciones de prueba. Complementos en instrucciones. Redirección local en ficheros de órdenes.

6) Robustez, depuración

- Robustez del shell-script: comprueba la inicialización de variables.
- Gestión avanzada de argumentos de línea de comandos (getopts).
- Pruebas sobre el tipo de variables (expr). La función eval.
- Gestión de señales.
- Nombrar y borrar archivos temporales.
- Caso especial de ejecución de un shell-script mediante cron (el crontab).
- Depuración de un shell-script: comandos de depuración.
- Señales de rastreo. Registro.
- Registro.

7) Extensiones Korn Shell y Bash

- Tablas de variables.
- Anotaciones específicas.
- Operaciones aritméticas.
- Alias rastreados.
- Noción de coprocesos.
- Comandos internos específicos.

8) Herramientas adicionales (grep, find, sed, awk)

- Herramientas que te ayudarán a crear scripts: grep y find.
- Expresiones regulares: herramienta grep.
- Búsqueda y tratamiento de archivos: herramienta de búsqueda.
- Sed: principios y aspectos de los scripts. Comandos sencillos de búsqueda, sustitución e inserción.
- Manipulación de flujos de texto con sed. Visión general de los comandos avanzados.
- Elementos generales de programación con awk. Uso de variables y funciones.
- Ejemplos completos de scripts awk (estadísticas del sistema, cálculos, etc.).
- Presentación de las funciones integradas de awk: matemáticas, tratamiento de cadenas, interacción con el sistema, etc.

FECHAS

Contacto