Administración de Sistemas Avanzada

Eduardo Grosclaude

2014-08-11

[V0.1 - Material en preparación, se ruega no imprimir mientras aparezca esta nota]

Resumen

En este escrito se presenta la descripción y material inicial de la asignatura **Administración de Sistemas Avanzada**, para la carrera de Tecnicatura Universitaria en Administración de Sistemas y Software Libre, de la Universidad Nacional del Comahue.

La materia es cuatrimestral en modalidad presencial y las clases son de carácter teórico-práctico, desarrolladas en forma colaborativa. Está preparada con los objetivos generales de capacitar al estudiante para **implementar configuraciones especiales de almacenamiento, aplicar programación avanzada a la automatización de tareas, y diseñar e implementar estrategias de respaldo y de tolerancia a fallos para servicios críticos**.

Página en blanco

Índice

Página en blanco

Parte I
La asignatura

1. Objetivos

De la carrera

Según el documento fundamental de la Tecnicatura, el Técnico Superior en Administración de Sistemas y Software Libre estará capacitado para:

- Desarrollar actividades de administración de infraestructura. Comprendiendo la administración de sistemas, redes y los distintos componentes que forman la infraestructura de tecnología de una institución, ya sea pública o privada.
- Aportar criterios básicos para la toma de decisiones relativas a la adopción de nuevas tecnologías libres.
- Desempeñarse como soporte técnico, solucionando problemas afines por medio de la comunicación con comunidades de Software Libre, empresas y desarrolladores de software.
- Realizar tareas de trabajo en modo colaborativo, intrínseco al uso de tecnologías libres.
- Comprender y adoptar el estado del arte local, nacional y regional en lo referente a implementación de tecnologías libres. Tanto en los aspectos técnicos como legales.

De la asignatura

- Saber implementar configuraciones especiales de almacenamiento
- Saber aplicar programación avanzada a la automatización de tareas
- Saber diseñar e implementar estrategias de respaldo
- Conocer formas de implementar estrategias de tolerancia a fallos para servicios críticos

2. Cursado

- Cuatrimestral de 16 semanas, 128 horas totales
- Clases teórico-prácticas presenciales
- Promocionable con trabajos prácticos

3. Contenidos

Contenidos mínimos

- Instalación sobre configuraciones de almacenamiento especiales.
- Scripting avanzado.
- Planificación de tareas.
- Virtualización.
- Alta Disponibilidad.

Programa

1. Scripting avanzado
   - Estructuras de programación
   - Scripting para tratamiento de archivos
   - Planificación de tareas
2. Configuraciones de almacenamiento
   - Arquitectura de E/S, Dispositivos de E/S, Filesystems
   - Diseños típicos de almacenamiento
   - Software RAID, instalación y mantenimiento niveles 0, 1, 10
   - LVM, instalación y mantenimiento
3. Estrategias de respaldo
   - Copiado y sincronización de archivos
   - Estrategias y herramientas de backup, LVM snapshots
   - Control de versiones
4. Virtualización
   - Formas de virtualización, herramientas. KVM, Proxmox, otras
   - Creación, instalación, migración de MV
   - Cloud. IaaS, PaaS, SaaS, etc.
5. Alta Disponibilidad
   - Clustering de LB, de HA, de HPC. Conceptos de HA.
   - Balance de Carga
   - Heartbeat, DRBD, Clustering de aplicaciones
   - Alta Disponibilidad en Redes. Bonding, STP

4. Bibliografía inicial

- Kemp, Juliet. Linux System Administration Recipes: A Problem-Solution Approach. Apress, 2009.
- Lakshman, Sarath. Linux Shell Scripting Cookbook Solve Real-World Shell Scripting Problems with over 110 Simple but Incredibly Effective Recipes. Birmingham, U.K.: Packt Pub., 2011.
- Parker, Steve. Shell Scripting Expert Recipes for Linux, Bash, and More. Hoboken, N.J.; Chichester: Wiley; John Wiley, 2011.
- Quigley, Ellie. UNIX Shells by Example. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

Parte II

Scripting Avanzado

1. Contenidos

1. Comandos básicos de archivos ls, cd, mkdir, cp, mv, rm, ln, patrones de nombres
2. Redirección y piping, comandos head, tail, more, less, grep
3. Variables, ambiente, aritmética
4. Sentencias de control if, for, while, case
5. Funciones
6. Arreglos
7. Expresiones regulares, uso de grep
8. Uso de sort, diff, comm, uniq, cut
9. Uso de cron
10. Otros intérpretes: sed, awk, Perl

2. Ejercitación básica

Redirección y piping

1. Crear un archivo conteniendo la salida del comando ls
2. Crear un archivo conteniendo la salida del comando ls -lR /tmp
3. Obtener las cinco primeras líneas del archivo anterior
4. Crear un archivo conteniendo las cinco primeras líneas y las cinco últimas del archivo generado en 2
5. Crear un archivo conteniendo las primeras cinco líneas de la salida del comando ls -lR /tmp
6. Usando el anterior, crear un archivo conteniendo esas líneas, numeradas
7. Crear un archivo conteniendo las últimas cinco líneas de la salida del comando ls -lR /tmp

Variables, ambiente

1. Asignar e imprimir el contenido de dos variables
2. Asignar dos variables, imprimir sus valores, intercambiar sus valores, imprimirlos
3. Crear un script que imprima un valor que será pasado como argumento
4. Crear un script que imprima dos valores que serán pasados como argumento
5. Crear un script que imprima todos los valores que le sean pasados como argumento

Sentencias de control

1. Imprimir cinco veces "Linux"
2. Imprimir cinco veces el contenido de una variable
3. Imprimir los números de 0 a 5
4. Imprimir los dígitos de -1 a 6
5. Imprimir los números de 0 a 99
6. Imprimir junto al nombre de cada archivo en el directorio actual, su tamaño y su fecha de modificación

7. Copiar los archivos terminados en .txt en archivos con igual nombre pero extensión .bak
8. Para cada archivo modificado hace más de cinco días en un directorio, mostrar su cantidad de líneas
9. Obtener mediante un cliente de HTTP una lista de archivos cuyos nombres están dados por una expresión variable y controlada por un lazo
10. De un conjunto de archivos tar, encontrar aquellas versiones de un archivo dado, contenido en ellos, que hayan sido modificadas entre dos fechas dadas.

Aritmética

```
$ declare -i num
$ num="hola"
$ echo $num
      0
$ num=5 + 5
      bash: +: command not found
$ num=5+5
$ echo $num
      10
$ num=4*6
$ echo $num
      24
$ num="4 * 6"
$ echo $num
      24
$ num=6.5
      bash: num: 6.5: syntax error in expression (remainder of expression is
          ".5")
$ i=5; j=$i+1; echo $j
$ i=5; let j=$i+1; echo $j
$ let i=5
$ let i=i+1
$ echo $i
      6
$ let "i = i + 2"
$ echo $i
      8
$ let "i+=1"
$ echo $i
      9
$ i=3
$ (( i+=4))
$ echo $i
      7
$ (( i=i-2 ))
$ echo $i
      5
$ let b=2#101; echo $b
$ let h=16#ABCD; echo $h
```

Arreglos

```
$ A=(1 2 3 cuatro cinco)
$ echo ${!A[*]}
0 1 2 3 4
$ echo ${A[4]}
cinco
$ echo ${A[*]}
1 2 3 cuatro cinco
$ A[2]='banana'
$ echo ${A[*]}
1 2 banana cuatro cinco
```

Arreglos asociativos

```
$ declare -A B
$ B=([francia]='paris' [espana]='madrid' [argentina]='buenos aires')
$ echo ${!B[*]}
espana argentina francia
$ echo ${B[*]}
madrid buenos aires paris
$ echo ${B[francia]}
paris
```

Here-Documents

```
$ cat > texto.txt << END
> Hola
> Probando...
> END
$ cat texto.txt
```

Traps

```
# man 7 signal
# 1 = SIGHUP (Hangup of controlling terminal or death of parent)
# 2 = SIGINT (Interrupted by the keyboard)
# 3 = SIGQUIT (Quit signal from keyboard)
# 6 = SIGABRT (Aborted by abort(3))
# 9 = SIGKILL (Sent a kill command)

trap limpieza 1 2 3 6 9

function limpieza
{
        echo "Recibimos senal - desmantelando..."
        # Borrar archivos temporarios
        rm -f ${tempfiles}
        echo Finalizando
}
```

3. Casos de uso

Investigar el sistema

1. Modificar la salida del comando blkid para conocer el UUID, el nombre y tipo, y punto de montado, de cada dispositivo de bloques del sistema.
2. Analizar archivos de log buscando conocimiento: duración de sesiones ssh por usuario, mensajes de mail entre usuarios, con histograma por tamaños, etc. (ver iptables.log, VI)
3. Detectar momentos en que la salida de vmstat muestra picos de I/O, procesos corriendo, procesos en espera, uso de swap, etc.

Recuperar espacio de almacenamiento

1. Encontrar los diez archivos más grandes en un directorio y sus hijos, imprimirlos junto con su tamaño de mayor a menor.
2. Encontrar los diez archivos más grandes en un directorio y sus hijos, moverlos a otro directorio (en otro filesystem).
3. Encontrar los diez archivos más grandes del sistema, imprimir el nombre de usuario dueño.
4. Agregar al script anterior el envío de notificación por mail al usuario responsable.
5. Encontrar archivos en directorios de usuario con la cadena "cache" en su nombre e imprimir el uso de disco de cada uno.
6. Idem, enviando nombres a un archivo y usándolo como lista para borrarlos, comprimirlos o moverlos.

Networking

1. Disparar un aviso cuando se pierde la conectividad a un conjunto dado de nodos de la red.
2. Analizar la salida del comando netstat para descubrir en qué momento aparece un nuevo port abierto y a qué aplicación corresponde.
3. Obtener un log de tráfico y obtener orígenes máximos y mínimos de tráfico, cantidades totales de bytes traficados por interfaz, etc.
4. Recoger estadísticas de espacio en disco, cantidad de procesos, carga de CPU, en diferentes nodos de la red, y centralizarlos en un nodo monitor que presente los resultados.

Seguridad

1. Detener el script si la identidad del proceso corresponde a root.
2. Solicitar información confidencial (como claves) con video inhibido.
3. Capturar señales para impedir la interrupción del script por BREAK o fallos de ejecución.
4. Utilizar MD5/SHAx para confirmar integridad de archivos.

Tratamiento de datos

1. Revisar el uso de los comandos cut, join, sort, uniq, comm.
2. Crear script que administra una base de datos en formato CSV.
3. Dado un archivo con una lista de direcciones IP, adjuntarles la resolución inversa de nombres correspondiente.
4. Crear un histograma de accesos por nombre de dominio, a partir de los paquetes registrados en un archivo de log generado por iptables.
5. Dada una base de datos CSV implementar búsqueda por expresiones regulares.

6. Dada una base de datos CSV implementar proyección sobre un conjunto de campos dados.
7. Convertir un listado de individuos PDF en archivo CSV.
8. Preparar un conjunto de scripts con un único punto de entrada para el administrador. Estos scripts mantendrán un conjunto de bases de datos en formato CSV:

```
alumnos: UID, Username, Apellido, Nombres, NoLegajo, Activo
materias: MID, Nombre, Carrera, Docente
cursadas: UID, MID, Ano, Cuatrimestre
```

   El dato Activo es booleano. Con estas bases de datos:

   - Listar todas las materias asignadas a un mismo docente.
   - Listar todas las materias cursadas por un alumno.
   - Listar todos los alumnos activos inscriptos en una materia.
   - Listar todos los alumnos que cursan una misma carrera dada durante un año dado.
   - Listar todos los alumnos, agrupados por materia cursada, dentro de cada año.
   - Listar todos los alumnos de un mismo docente.
   - Dado un alumno por su legajo, consultar su estado Activo/Inactivo.
   - Para aquellos alumnos que hace más de tres años que no se inscriben en ninguna cursada, pasar su dato Activo a falso (Inactivo).
   - Generar un par de archivos en el formato de /etc/passwd y /etc/shadow para todos los alumnos activos.
   - Generar un directorio /home/usuario para cada alumno activo, con UID correspondiente.

Accesibilidad para usuarios finales

1. Preparar un script con interfaz gráfica para copiar archivos seleccionados a una carpeta preestablecida con el fin de obtener un backup periódico de todos sus contenidos.
2. Preparar un script con interfaz gráfica que presente los cinco directorios con mayor ocupación de almacenamiento dentro del home del usuario.
3. Agregar interfaz gráfica a los scripts de administración de bases de datos de alumnos y materias.

Parte III

Estrategias de Respaldo

Parte IV

Virtualización

Parte V

Alta Disponibilidad

Parte VI
Anexos

iptables.log

```
Logged 539 packets on interface eth1
  From 0000:0000:1011:1213:0100:0000:0000:0000 - 3 packets to icmpv6(130)
  From 0000:0000:0000:859e:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0023:ff53:4d42:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:3433:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:3132:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:7d3a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:6e63:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:937f:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0000:0000:0000:0000:0000:0000:0000:0000 - 2 packets to icmpv6(130)
  From 0000:0000:0000:0000:0100:0000:0000:0000 - 40 packets to icmpv6(130)
  From 0000:0000:0000:6569:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 000e:175f:531c:580e:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0011:11db:a2d4:0a00:0100:0000:0000:0000 - 2 packets to icmpv6(130)
  From 002c:799a:0694:8c26:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0100:0000:0600:0000:0100:0000:0000:0000 - 2 packets to icmpv6(130)
  From 0101:080a:00c6:0621:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:00c6:f565:0007:994d:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:00c7:39ad:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:00c7:3e49:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:00c7:5bd1:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:1551:7183:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:1a9a:1543:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4553:94db:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4557:126c:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4559:6ffd:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4559:e9ac:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455a:3fd8:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455a:cc78:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455c:c658:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455d:4147:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455d:bbfc:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:455e:3351:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4567:104c:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4575:8fa3:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4575:fcd3:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4584:d388:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:458c:f368:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0101:080a:4594:8809:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0102:0417:0000:0000:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0102:a16d:0a00:00c8:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 0204:05b4:0402:080a:0100:0000:0000:0000 - 2 packets to icmpv6(130)
  From 0300:0000:0400:0000:0100:0000:0000:0000 - 3 packets to icmpv6(130)
  From 036b:696d:0675:6e63:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 10d4:d5cb:f80c:14d5:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 1400:0300:9709:0000:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 2269:6422:3a20:2232:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3037:3337:3431:3832:0100:0000:0000:0000 - 1 packet to icmpv6(130)
```

```
  From 3135:3332:3a38:3439:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3234:3238:323a:3834:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3333:3238:323a:3834:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3335:3839:323a:3834:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3336:3532:323a:3834:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3336:3837:3039:3132:0100:0000:0000:0000 - 8 packets to icmpv6(130)
  From 3430:3734:3330:3532:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3432:3230:3239:3a33:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3432:3635:3239:3a33:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3433:3230:3332:3a38:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3433:3534:3039:3a33:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3237:3339:313a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3330:3238:313a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3330:3636:313a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3330:3930:323a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3334:3533:313a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 3734:3334:3736:393a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 616a:6f72:223a:2031:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 6d65:6e74:7322:3a7b:0100:0000:0000:0000 - 9 packets to icmpv6(130)
  From 6f20:3134:3037:3433:0100:0000:0000:0000 - 2 packets to icmpv6(130)
  From 7473:223a:7b7d:7d00:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 7473:223a:7b7d:7d32:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From 7473:223a:7b7d:7d3a:0100:0000:0000:0000 - 4 packets to icmpv6(130)
  From 7473:223a:7b7d:7d7b:0100:0000:0000:0000 - 3 packets to icmpv6(130)
  From 756e:6e69:6e67:223a:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From bf1d:f661:984e:fcb0:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From c011:fdda:43fe:4d59:65fe:e1aa:c7d5:683a - 1 packet to icmpv6(130)
  From c012:aa5c:173c:261c:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From c012:cfa3:e641:3b5f:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From c013:4b15:1c15:3311:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From c013:9389:bcf8:f7d4:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From c014:ff7d:0000:0000:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From e063:837f:0000:0000:0100:0000:0000:0000 - 3 packets to icmpv6(130)
  From e063:837f:2077:937f:0100:0000:0000:0000 - 1 packet to icmpv6(130)
  From e063:837f:301f:937f:0100:0000:0000:0000 - 3 packets to icmpv6(130)
  From 0.0.0.0 - 154 packets to igmp(0)
  From 10.0.3.4 - 154 packets to igmp(0)
  From 10.0.3.21 - 75 packets to igmp(0)
  From 10.0.3.209 - 2 packets to igmp(0)

Listed by source hosts:
Logged 18 packets on interface virbr0
  From fe80:0000:0000:0000:5054:00ff:feed:8246 - 18 packets to udp(5353)

Listed by source hosts:
Logged 50 packets on interface wlan0
  From 10.0.4.1 - 2 packets to udp(68)
  From 64.233.186.188 - 15 packets to tcp(44421,53418)
  From 64.235.151.8 - 5 packets to tcp(56214)
  From 173.194.42.0 - 1 packet to tcp(56677)
  From 173.194.42.21 - 3 packets to tcp(58597)
  From 173.194.42.22 - 7 packets to tcp(35536)
  From 173.194.42.75 - 3 packets to tcp(48585)
  From 173.194.42.85 - 4 packets to tcp(44550)
  From 173.194.42.86 - 1 packet to tcp(51940)
```

```
From 192.168.1.1 - 2 packets to udp(68)
From 192.168.2.1 - 2 packets to udp(68)
From 195.135.221.134 - 1 packet to tcp(51756)
From 195.154.174.66 - 1 packet to tcp(58047)
From 200.42.136.212 - 3 packets to tcp(59351,59361)
```