IV. REALIZACIÓN

Una de las mejores maneras para comprender la IN y para conocer la problemática, es participar en la instalación manual de una plataforma IN.

4.1 Instalación en "base arrière" de una plataforma

4.1.1 ¿ Por qué la instalación en "base arrière"?

El desarrollo de una instalación cubre las siguientes actividades:

- Identificación de las obligaciones de la instalación.
- Preparación de la instalación.
- Instalación y puesta en marcha.

Las plataformas IN están preinstaladas en las fábricas de HP en Escocia siguiendo los documentos Alcatel (fase de industrialización). Desde la llegada de éstas a Alcatel, un instalador se hace cargo de ellas para llevar a cabo una instalación completa (excepto el servicio) teniendo en cuenta los requisitos del cliente. Esta instalación en el emplazamiento de Alcatel es un nuevo proceso de instalación llamado en "base arrière" que permite:

- Limitación de los riesgos de instalaciones incorrectas y mejores controles.
- Disponibilidad de los expertos en "base arrière":

  - Ingeniero RTM (MAP)
  - Bomberos
  - HP

- Intervenciones HP más rápidas y más eficaces.
- Mejor imagen de la marca frente al cliente.

Y es por todas estas razones que se decidió instalar las plataformas en "base arrière", en Vélizy, en Francia.

4.1.2 Procedimiento general

Una instalación de plataforma IN corresponde a una configuración material y software del servidor. Todos los servidores utilizados para tratar la IN son productos HP de enorme potencia (4 CPUs, hasta 32 Go de RAM.). Cada servidor funciona bajo el sistema operativo Tru64 v4.0F.

Estas instalaciones responden a un proceso de calidad.

Así pues, cada instalador sigue los procedimientos definidos en el "Tome Volume". Este documento interno a Alcatel, sinónimo de saber hacer, incluye entre otros las cuatro etapas siguientes:

1. Comprobación de los diferentes elementos materiales que componen el servidor: discos duros, la tarjeta red, conexión Ethernet.

2. Configuración de los servidores: cambio de las direcciones de los servidores en función de las de la red TCP/IP del cliente, reglado de la hora local, sincronización de los servidores y modificación de ciertos ficheros de configuración.

3. Instalación de los llamados subconjuntos de la plataforma IN, con el fin de poder utilizar los diferentes elementos de los servicios IN como "Comprov" (gestor de la plataforma) y "Accman" (gestor del operador).

4. Verificación e instalación del software: comprobación de las licencias y de los parches, de la configuración del núcleo UNIX y de la configuración de los discos.

Este proceso de calidad permite al servicio MAP saber con precisión el estado de la instalación en cada momento y en el peor de los casos poder intervenir cuanto antes, si un problema surge durante la instalación. Tiene por objeto garantizar la fiabilidad de la instalación y asegurar el seguimiento de todas las acciones realizadas sobre la plataforma.

4.1.3 Su papel en el proyecto

La comprensión de la instalación de una plataforma IN requiere un estudio detallado del "Tome Volume", pero también de la práctica.

Pude entonces participar durante dos semanas en la instalación de la plataforma para un cliente de Camboya descubriendo, en paralelo, el "Tome Volume". El instalador tenía por misión indicarme los procesos que le aparecían más repetitivos o que eran fuentes de error.

Esta fase de instalación y de análisis fue esencial para la realización del proyecto. Me permitió conocer la importancia de cumplir los *principios de automatización (1 y 3).*

Además, pude familiarizarme con el sistema UNIX (sistema operativo de los servidores) y con las Redes Inteligentes. Aprendí igualmente el papel de la IN: la implementación general de una plataforma IN en las Redes de telecomunicaciones existentes.

Otra contribución de esta fase a mi proyecto fue la utilización de la herramienta ya realizada para la automatización: el generador de ficheros sistemas.

4.1.4 Descripción o detalles de la instalación

La plataforma instalada, destinada a un cliente de Camboya, es una FULL IN monolítica compuesta de dos servidores SMP y de dos servidores SCP (FEP + BEP). Es una plataforma R2.2.05. Una vez llegada a Vélizy, fue montada (conectando entre si sus elementos) y conectada a la red eléctrica.

Comenzamos comprobando el nivel físico y la configuración del sistema.

A continuación, sincronizamos a los servidores, configuramos las interfaces y modificamos los ficheros sistema */etc/hosts* y */etc/routes*, ficheros en los cuales son declarados las direcciones IP y las rutas, así como todos los ficheros que hacen referencia a los nombres de los servidores.

Proseguimos por una comprobación de la conectividad con la red, del estado de los discos duros, del sistema de ficheros y las CPUs, y una comprobación de las interfaces SS7.

Después de esto, instalamos el "middleware" que es la interfaz entre la capa material y la capa de aplicación. Incluye diferentes niveles que es necesario instalar uno tras otro: el nivel 004, el nivel 027 y el nivel 115. Cada nivel integra los añadidos ("addendums") del nivel anterior y que son la corrección de los errores descubiertos en el nivel precedente además de las nuevas funcionalidades.

Continuamos configurando los puestos de explotación que sirven para administrar la plataforma visualizando las alarmas eventuales gracias a la aplicación llamada "Alarm Box" que permite señalar un mal funcionamiento de la plataforma: cuando hay alarmas la interfaz gráfica es roja si no es verde.

Por último, verificamos el software instalado, las licencias que corresponden a dicho software y los parches presentes en cada servidor, igualmente los datos del núcleo y la configuración de los discos duros.

Después de estas comprobaciones, el RTM responsable de la plataforma se aseguró que los niveles hardware, sistema, "middleware" y base de datos Oracle estaban correctamente instalados y configurados. Esta comprobación es justamente una de las partes de la instalación de las plataformas que se quiere automatizar debido a su repetitividad y a su importancia.

Cuando todo esto se realizó, el instalador lanzó un programa, el "Webes", que memoriza en un fichero el estado de la plataforma antes de su salida hacia la ubicación del cliente. El RTM recuperó estos ficheros y los puso en el sitio Intranet del MAP para permitir el control y seguimiento de la plataforma.

En la figura siguiente, podemos observar el estado en el cual se encuentra la plataforma IN después del proceso de instalación hecho en Vélizy:

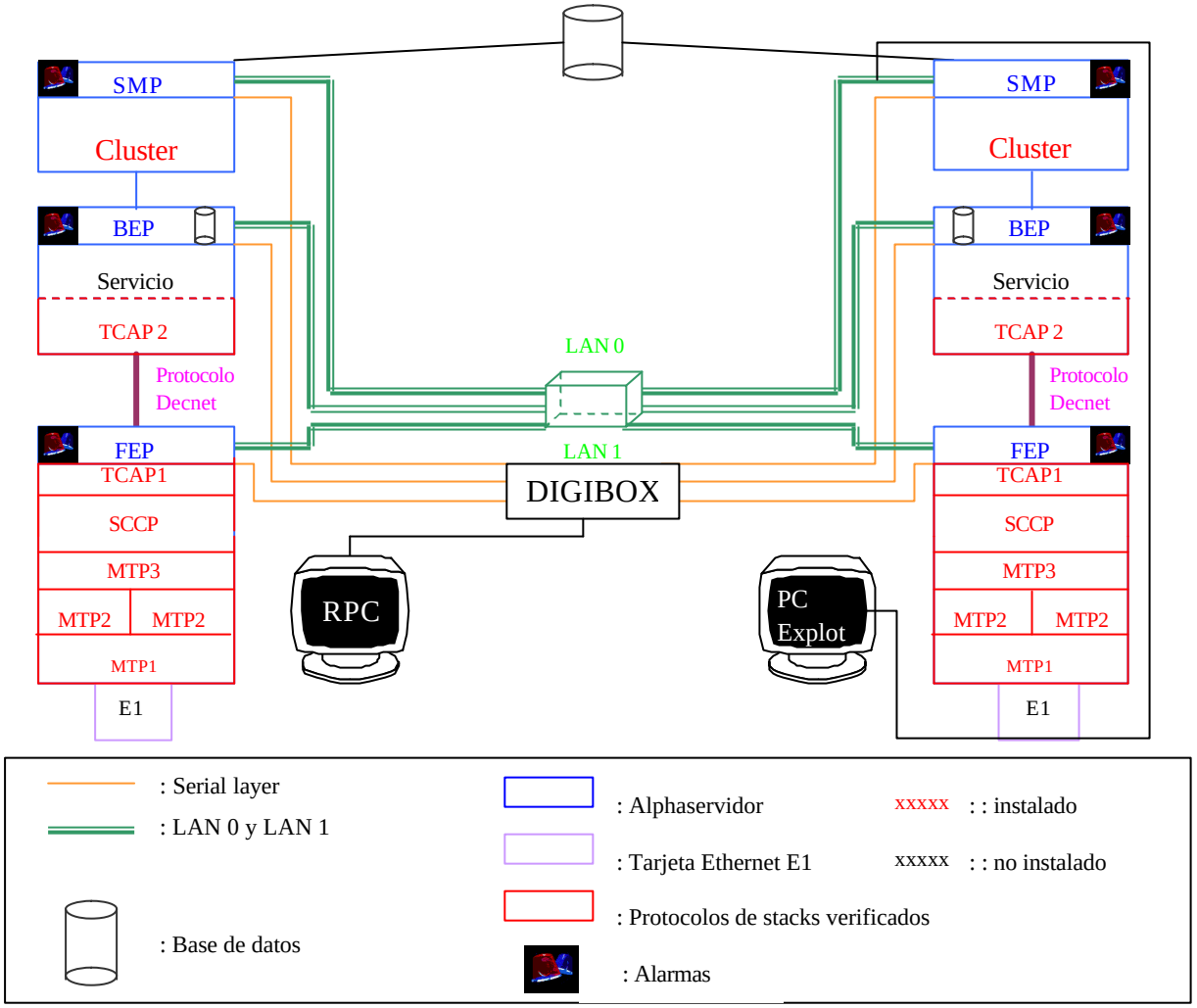

Figura 19: Plataforma IN después de la instalación en “base arrière”

Sólo aclarar que el PC Remoto o también llamado RPC es un ordenador distante que permite acceder a todos los elementos de la plataforma mediante un equipo llamado “Digibox”. Esto permite controlar los parámetros de configuración de los servidores y de los interruptores.

4.1.5 Enseñanzas

La instalación de esta plataforma me fue muy beneficiosa para delimitar las dificultades y definir mejor las partes automatizables. Constaté que una instalación exige mucha concentración ya que es necesario, por una lado, distinguir las manipulaciones que hay que hacer según el tipo de servidor (SMP o SCP), y, por otro lado, no olvidar ninguna etapa, lo que podría requerir una vuelta atrás en la instalación, incluso una desinstalación completa.

4.2 Automatización del proceso de verificación de plataformas IN

La verificación tiene como principal objetivo asegurar que el conjunto de los procedimientos de configuración ha sido aplicado correctamente sobre la plataforma.

En general, no se tiene mucho tiempo para verificar una plataforma, por esta razón la utilización de scripts debe ser rápida y eficaz. De hecho, la verificación debe ser realizada en un intervalo de tiempo comprendido entre medio día y dos días. Existen diferentes situaciones en las cuales se quiere o se debe realizar una verificación:

- Después de la instalación en "base arrière", una comprobación completa (hardware y software) de la plataforma es llevada a cabo antes de su entrega al cliente. Permite asegurar que los niveles hardware, sistema, "middleware" y la base de datos han sido bien instalados.

- Una vez llegada la plataforma a la ubicación del cliente y antes de continuar con la instalación de los servicios, el instalador la verifica una vez más para comprobar que la información recibida sobre su nivel de instalación, después de Vélizy, es correcta.

- Después de la instalación de servicios in situ.

- Antes de hacer una intervención (una extensión de memoria o añadir discos duros, por ejemplo) sobre una plataforma en funcionamiento. Hay que hacer una verificación de la plataforma ya instalada y utilizada por el cliente con el fin de conocer su estado antes de la intervención.

- Reutilización de plataformas clientes: recuperamos antiguas plataformas del cliente para hacer migraciones de su versión actual a una más moderna o para cambiarlas de configuración, por ejemplo. Antes de trabajar en ellas, las verificamos por la misma razón.

Se puede entonces percibir la importancia que tiene el control de la plataforma IN así como el tiempo necesario para hacerlo. De estas razones deriva el interés de automatizar la verificación de plataformas instaladas.

Puedo ahora numerar los diferentes objetivos que pretendo alcanzar con la verificación automática:

- Aumento de la fiabilidad de la instalación tanto para el instalador como para el miembro del servicio MAP (RTM).

- Tener un rastro del estado completo de la plataforma, muy útil para los RTM.

- Ahorro de tiempo.

4.2.1 Situación a mi llegada

Este proyecto de automatización vio la luz antes del principio de mis prácticas. Por ello, antes de continuar con mi trabajo personal y su contribución al proyecto global, es interesante comentar el punto en el que se encontraba dicho proyecto de automatización cuando llegué al servicio MAP.

Mi predecesor había realizado una aplicación en Visual Basic que generaba automáticamente una serie de *ficheros sistema* de la plataforma antes de la propia instalación. Explicaré detalladamente esta herramienta mas tarde, en el apartado dedicado a la instalación ya que se corresponde mejor con esta parte del proyecto.

Igualmente existía un script que creaba los llamados *ficheros de configuración*. Y estos ficheros sí que son útiles para la automatización de la verificación. No son ficheros propios de la plataforma pero contienen los datos de la misma de tipo: número y nombre de servidores, versión de la plataforma y del "firmware", direcciones IP, etc. (para una descripción mas detallada del formato del fichero de configuración remítase al **Anexo A**).

El fichero de configuración, llamado **conf_file_<nom-servidor>**, es específico a cada servidor ya que los datos que contienen no son exactamente iguales dependiendo por ejemplo del tipo de servidor. Por lo tanto, habrá uno por cada servidor de la plataforma.

Conociendo el formato general del fichero de configuración, también pueden ser rellenados y creados a mano en el caso de que la instalación no se haga automáticamente aunque la verificación sí. Y deben ser creados consultando los documentos CNI (Customer Network Information) y PMD (Projet Manufacturing Documentation).

Veremos posteriormente su utilidad en la verificación automática.

4.2.2 Especificaciones

Para la realización de los scripts de verificación de plataformas IN se tuvo en cuenta las siguientes especificaciones:

- Se trata de una verificación hardware y software de la plataforma IN R2.2.05. Para ello, los scripts deben realizar todas las instrucciones o comandos UNIX de los documentos:

  - **Check List System**: esta compuesto por numerosos tests cada uno de los cuales sirve para verificar una parte del sistema total. Así pues, permite verificar el estado de lo que se llama sistema de la plataforma, es decir, las licencias, los subconjuntos instalados, la configuración de red, el estado de los discos, la memoria física, los valores del Kernel …
  - **Check List Middleware** : de la misma forma, permite verificar la buena instalación del "middleware", es decir, la base de datos Oracle, la detección de las alarmas, el "Netwatcher", las direcciones de los nodos…
  - **NITL9**, es una nota de información técnica y logística: permite hacer las últimas modificaciones y los últimos tests de verificación sobre la plataforma.

- Incluir un script destinado a la automatización de la verificación del nivel de "addendum" de la plataforma así como un script para verificar la "stack" instalada.

- Cumplir los principios 1, 2 y 3 de automatización.

- Existencia de dos versiones o métodos de utilización de los scripts: una versión detallada destinada a los instaladores para que participen en el control y una versión más rápida para los RTM. Ambas versiones serán realizadas a partir del mismo código fuente.

- Utilización del los ficheros existentes: los ficheros de configuración generados o no automáticamente antes de la instalación serán muy útiles.

- Estos scripts deben ser reutilizables y transportables. Además habrá que asegurar la viabilidad de la utilización de estos programas a distancia en plataformas en funcionamiento.

- Consideración de todas las posibles configuraciones de plataforma IN Alcatel R2.2.05: los scripts deben ser aplicables a cualquier tipo de servidor.

- Utilización rápida y simple.

- La versión de Shell utilizada para la escritura de scripts será el Shell de Bourne por ser compatible con todas las versiones de UNIX, sistema operativo de Alcatel.

- Los resultados del control automático deben ser indicados por pantalla pero serán igualmente destinados a unos ficheros de resultados que podrán ser recuperados y utilizados posteriormente a conveniencia.

- La lengua utilizada debe ser el inglés, lengua oficial de Alcatel. Por lo tanto todos los comentarios, cuestiones y mensajes son escritos en inglés, permitiéndose así una utilización internacional de los scripts.

4.2.3 Elaboración

Después de haber instalado manualmente una plataforma IN y de haber realizado y comprendido el procedimiento de control y validación post-instalación, comencé a desarrollar los diferentes scripts de verificación.

4.2.3.1 Concepción, desarrollo y primer test

La verificación automática consiste en comparar los datos encontrados en el propio servidor, a través de los comandos UNIX de los documentos que queremos automatizar, con los datos que hay que tener teóricamente después de la instalación. Por lo tanto, los scripts necesitan que se les suministren de una serie de parámetros a lo largo de su funcionamiento.

El procedimiento que he seguido para su realización es un método progresivo:

- **Definición de las partes interesantes a automatizar**: los documentos están formados por varias partes llamadas tests y no todas son factibles con los scripts o no todas suponen una verdadera ganancia de tiempo.

- **Reparto de estas partes en varias etapas**: conjuntos de partes o tests son reagrupados en programas diferentes dependiendo de lo que verifican. Por lo tanto un solo script consta de varios tests o partes.

- **Desarrollo y test de cada etapa o script**.

- **Validación y entrega de cada script**: después de cada test de un script, si el resultado es favorable, se valida y se entrega, es decir, se introduce en el servidor del MAP, poniéndose a disposición de todo el servicio.

Así, fui escribiendo y probando los scripts progresivamente. Cada vez que una parte era probada y validada la entregaba para su utilización en espera de ser completada.

Todos los scripts fueron sometidos a un primer test sobre la plataforma "KIM" del servicio MAP que es una STEP IN Duplex destinada a los ensayos de scripts y validación de procedimientos. Asegurándome al menos que dichos scripts no contenían errores de sintaxis y que eran válidos para una configuración de este tipo.

Siguiendo este procedimiento, llegué a tres programas principales y a una serie de rutinas o scripts secundarios que son llamadas por dichos programas:

- **check_system.sh:** tiene dos posibles métodos de utilización y está compuesto a su vez por múltiples tests realizados a través de llamadas a los scripts:
  - capa.sh
  - crontab.sh
  - date.sh
  - dynamic.sh
  - fichier.sh
  - firmware.sh
  - instal.sh
  - level004.sh
  - level027.sh
  - level115.sh
  - servers.sh

- **nitl9.sh:** realiza igualmente una serie de tests de verificación pero sin llamar a subrutinas, sino un test tras otro, pidiendo información cuando es necesario.

- **check_middleware.sh:** permite la posibilidad de dos modos de utilización. Los scripts que son llamados por dicho programa principal son:

  - additional.sh
  - connexion.sh
  - general.sh
  - oracle.sh
  - smp.sh
  - ss7.sh

Estos scripts dan lugar a una serie de ficheros de resultados que son generados durante la ejecución de los programas:

Nombre script	Ficheros de resultados
check_system.sh	system_verif filesystem Licences
check_middleware.sh	middle_verif filesystem ss7_verif
nitl9.sh	nitl9_file

Figura 20 : Tabla de ficheros de resultados

Después de cada control, estos ficheros serán analizados para encontrar los posibles errores de instalación. Para una descripción detallada de los contenidos de dichos ficheros, remítase al **Anexo A**.

4.2.3.2 Modos de empleo

En el caso de los scripts principales: check_system.sh y check_middleware.sh creé dos versiones o modos de empleo:

- **La versión instalador**: versión que hace participar mas al operador, instalador.
- **La versión RTM**: versión más rápida y que requiere menor intervención humana.

Para realizar un único código común a ambas versiones, introduje una cuestión al principio de la ejecución del script principal preguntando si se trata de un instalador o de un RTM. Si la respuesta es afirmativa (versión instalador) entonces un fichero vacío llamado *installateur* es creado en el directorio de trabajo (temporal).

Antes de cada test, utilizo el fichero vacío para conocer la versión en la cual estoy. A partir de ese punto, el código fuente es el mismo para las dos versiones y las dos únicas diferencias entre los dos métodos de utilización de scripts de verificación son:

- El modo de obtener los parámetros necesarios para realizar las comparaciones, es decir, la verificación automática. Si el fichero vacío (*installateur*) existe, se pide al instalador que introduzca los datos, si no se recuperan automáticamente del fichero de configuración conf_file.

- La pregunta, delante de cada test, para continuar con el test siguiente. Únicamente existente en la versión instalador

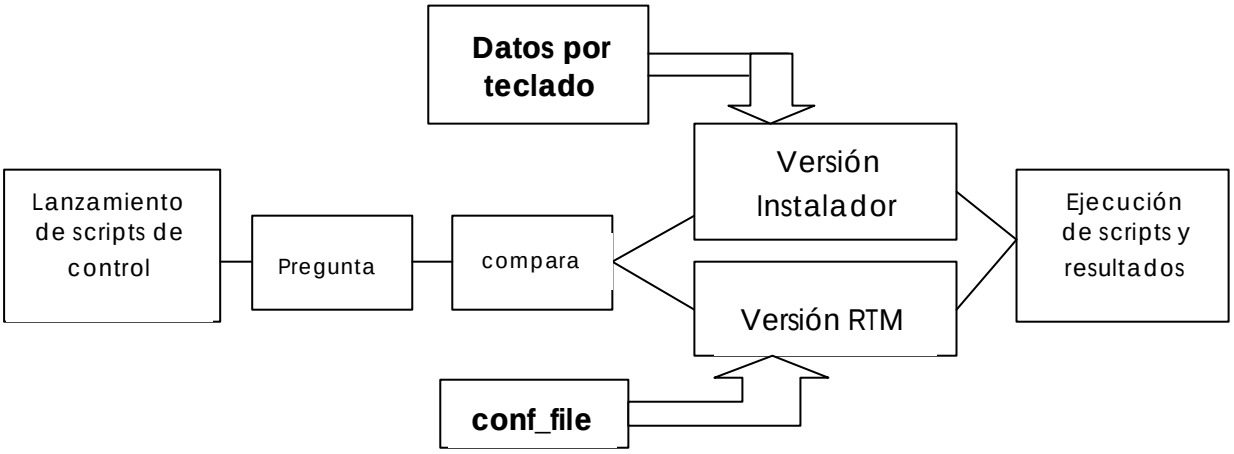

Figura 21: Obtención de datos para las dos versiones

Todas las pruebas y todos los scripts son entonces iguales para ambas versiones.

4.2.3.3 Métodos de utilización y ejecución de scripts

Puedo ya explicar mas detalladamente, con ayuda de gráficos, cómo realizar la verificación automática de una plataforma IN con los scripts y cómo se van ejecutando los diferentes scripts, así veremos mas claramente las diferencias existentes entre las dos versiones.

El procedimiento que permite utilizar los scripts de control está formado por diferentes etapas expuestas en el siguiente esquema:

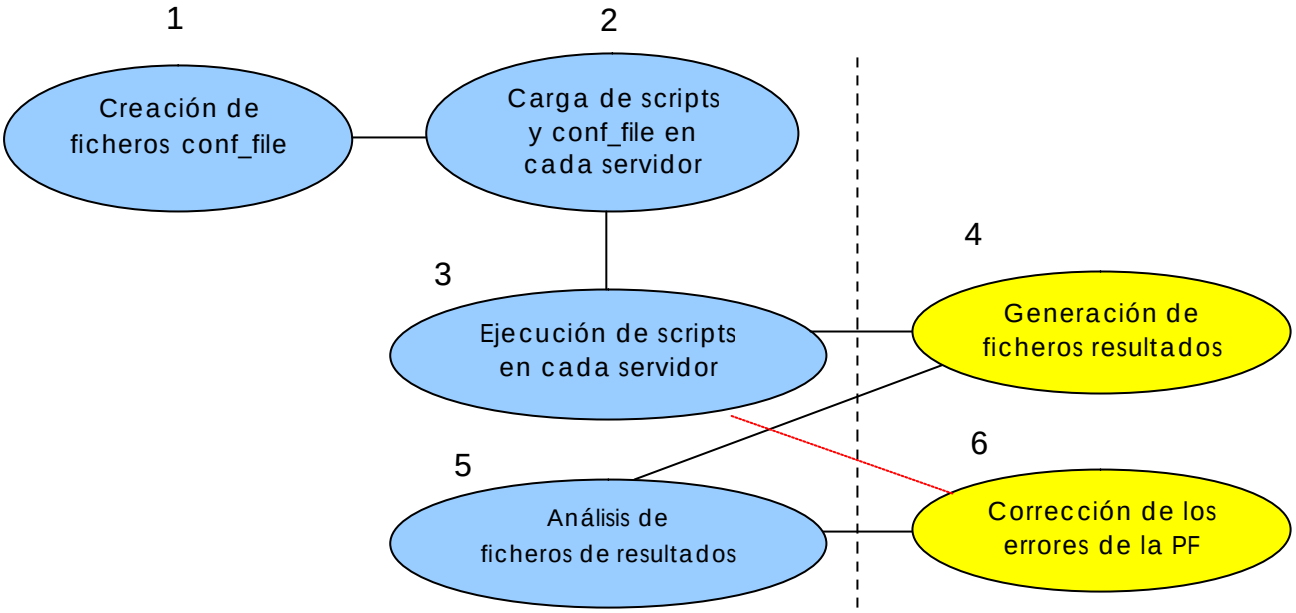

Figura 22: Método seguido para la verificación automática de plataformas

1. El fichero de configuración es esencial para la versión RTM de scripts. Si la plataforma ha sido instalada automáticamente, estos ficheros son creados automáticamente anteriormente (veremos esto más tarde). Si no, serán completados a mano antes de la verificación.

2. Hay que tener los scripts en cada servidor. Lo normal es que estemos obligados a pasar por el RPC para hacer la transferencia de ficheros del disquete a los servidores.

   Primero montamos el disquete en el RPC y metemos los ficheros y los scripts en uno de sus directorios. Luego los trasladamos del RPC a los servidores por FTP. Este punto está explicado gráficamente en el **Anexo A**.

3. Ejecutamos los scripts. Etapa que detallaré en el siguiente apartado.

4. La ejecución de los scripts genera un conjunto de ficheros de resultados.

5. Los ficheros de resultados son analizados manualmente. En caso de error en la instalación de la plataforma, podemos saber fácilmente donde se encuentra el error mirando el mensaje de error en estos ficheros.

   Los errores encontrados son debidos normalmente a la modificación de los ficheros de la plataforma realizadas por los instaladores durante la instalación con el editor *vi*. Los resultados del control son también señalados al operador sobre la pantalla.

6. Por último, se corrigen los errores en el caso de que los haya. Y para verificar que la corrección ha sido adecuada, se vuelven a lanzar los scripts si es necesario.

Situándonos ahora en la etapa número 3 del procedimiento anterior, es decir, con todos los scripts y ficheros necesarios en cada servidor, procedemos a ejecutarlos, realizando los siguientes pasos:

1. Abrimos una sesión en cada uno de los servidores de la plataforma:

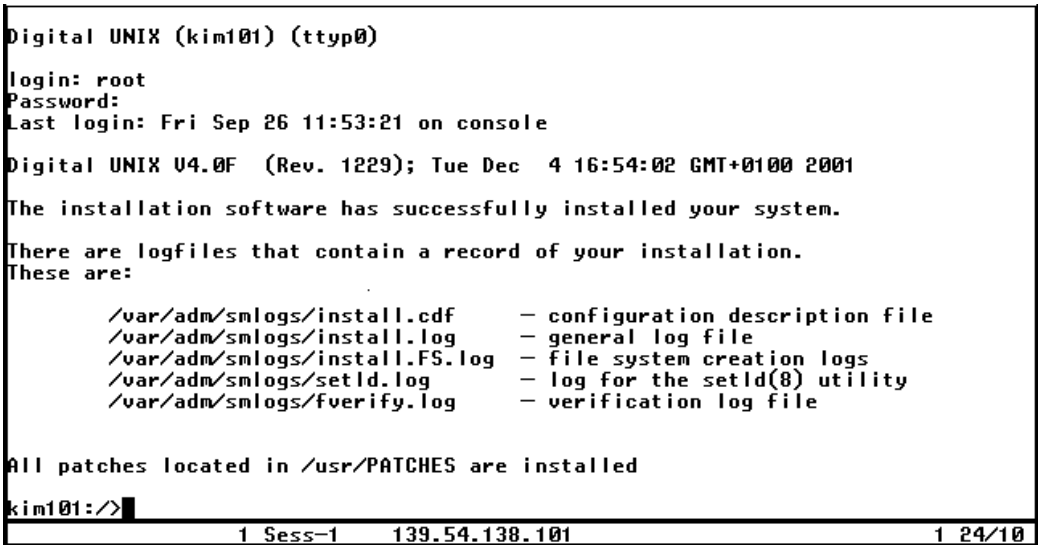

Figura 23: Pantalla de apertura de sesión

2. Nos situamos en el directorio de trabajo recomendado (temporal). Luego, lanzamos uno de los scripts principales:

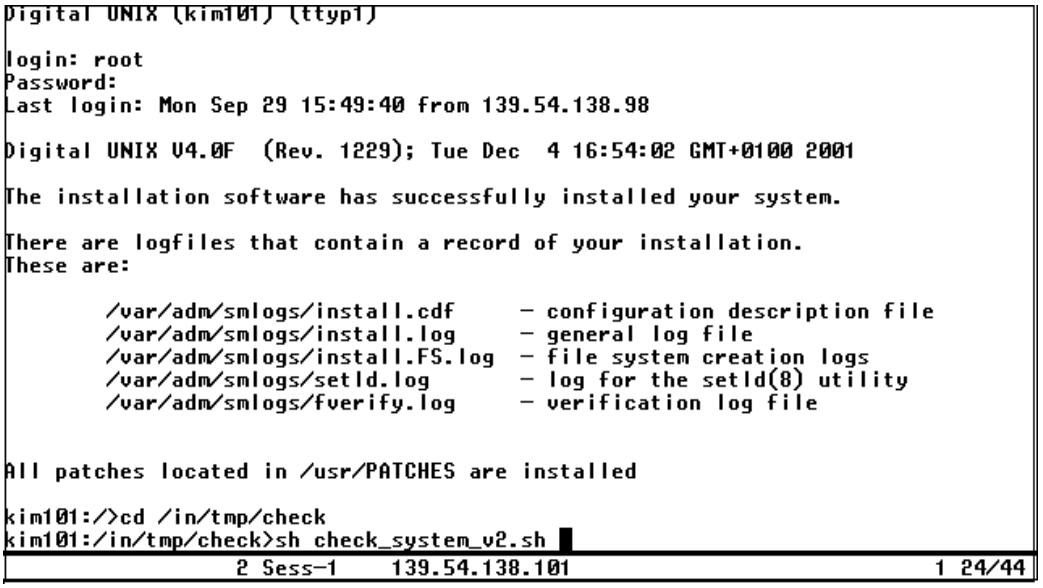

Figura 24: Ejecución de un script principal

3. Las primeras preguntas que aparecen son la del tipo del servidor sobre el cual se ejecuta el programa y la de la versión que queremos usar:

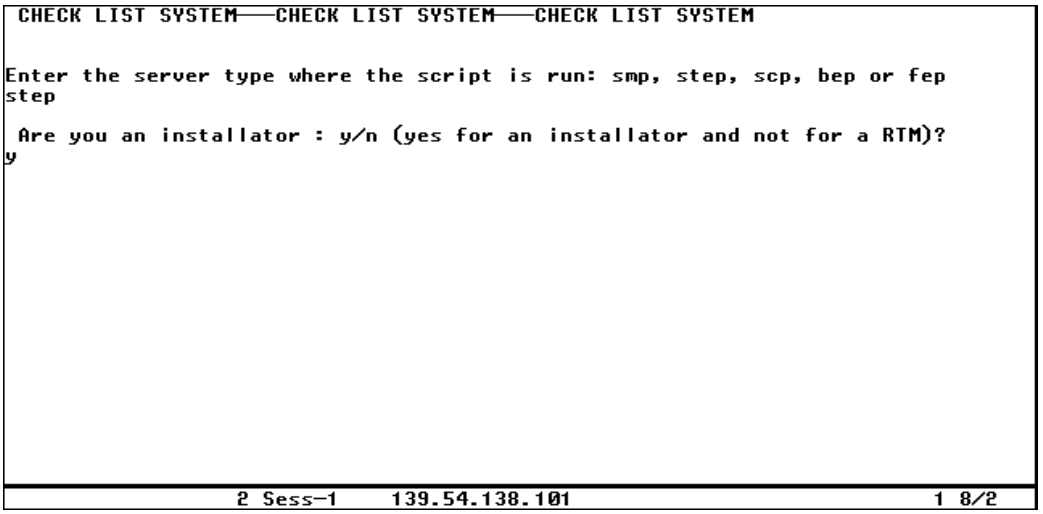

Figura 25: Cuestiones del tipo de servidor y modo de empleo

De esta forma, sabremos que test debemos o no hacer en cada caso ya que no todos los tests de un script son realizados para todos los tipos de servidores. Así con una simple comprobación de la variable que contiene el tipo de servidor justo antes del test estará solucionado el problema. Por este motivo es importante probar el conjunto de todos los scripts sobre todas las posibles configuraciones ya que sólo de esta forma estaremos seguros de su correcto y total funcionamiento.

Además los programas están preparados para posibles errores humanos del operador. De esta forma, si se introduce un dato que no es válido porque no existe o no es un valor posible para el parámetro demandado, obtenemos por ejemplo:

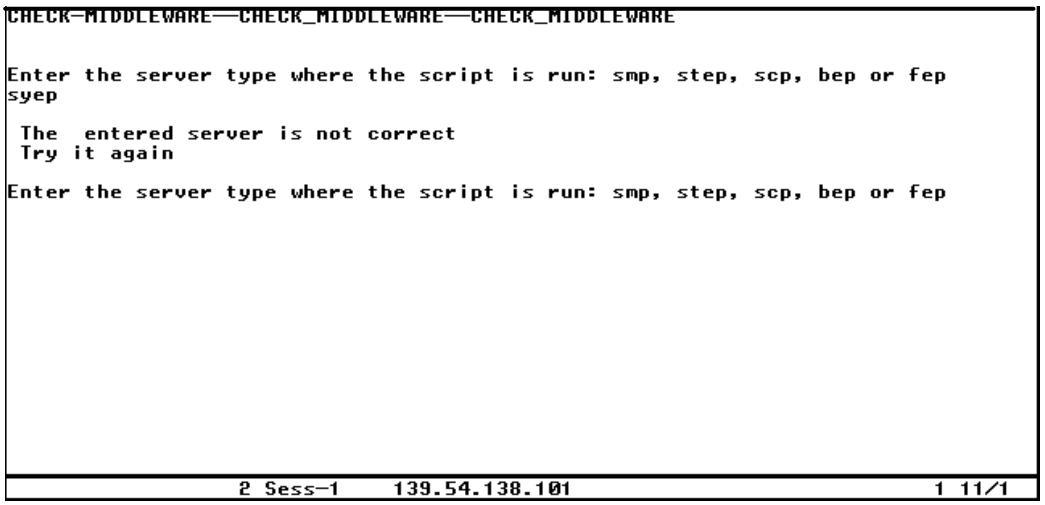

Figura 26: Error del operador

4. **A:** Si la versión elegida es la del instalador, aparece un menú con todas las opciones que podemos elegir.

En el caso del script principal **check_system.sh**, el menú es:

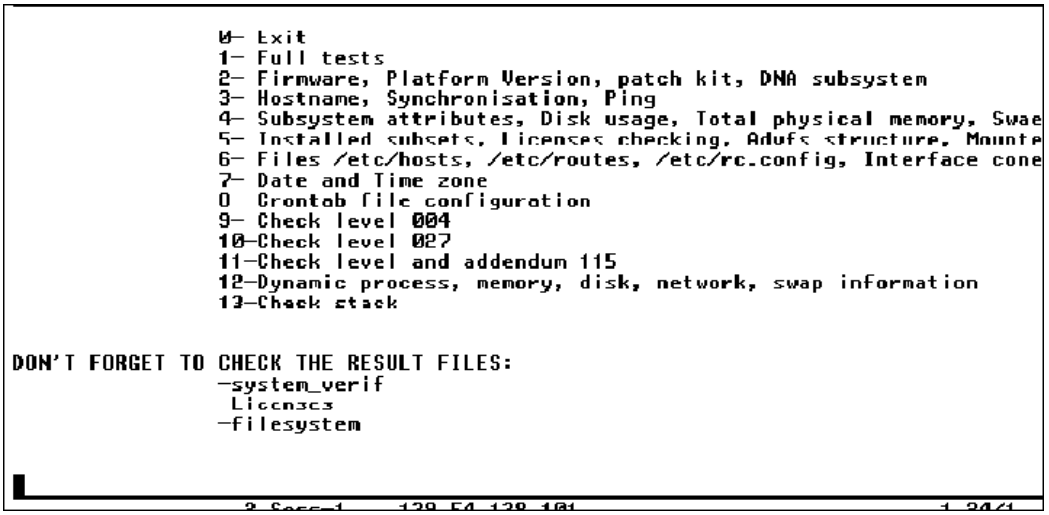

Figura 27: Menú del script principal check_system.sh

Y en el caso del script principal **check_middleware.sh**:

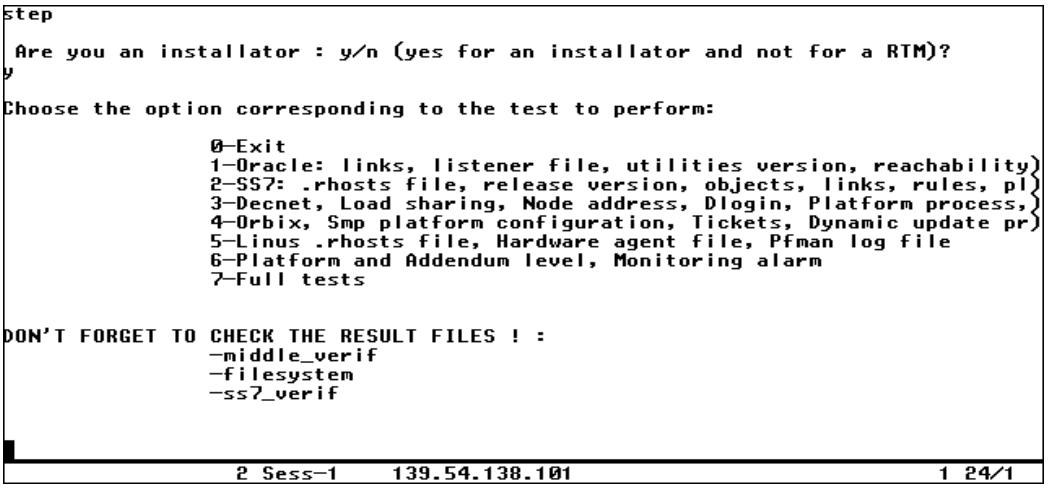

Figura 28: Menú del script principal check_middelwa re.sh

Cada opción en el menú se corresponde con una subrutina, de esta forma podemos escoger los tests que queremos pasar, es decir, las scripts a ejecutar.

Existe también la opción para realizar todos los tests de una vez (*1-Full tests*), lo cual supondrá la ejecución de todas las rutinas o scripts seguidos, uno tras otro.

5. **B:** Si, en cambio, la versión elegida es la de RTM, entonces al contestar a la pregunta negativamente y pulsar la tecla <enter>, todos los tests empiezan a ejecutarse.

   En este caso no se pide introducir ningún parámetro a través del teclado ya que éstos son tomados automáticamente del fichero de configuración correspondiente. Pero podemos ver igualmente por pantalla los nombres de los scripts y de los tests que se realizan en cada momento y el resultado de dichos tests.

Una vez todos los tests han sido terminados, aparece el mensaje recordando los ficheros de resultados que hay que consultar.

6. Vamos a suponer que continuamos en la versión instalador y elegimos una opción del menú. Si la opción elegida es la de realizar de una vez todos los tests:

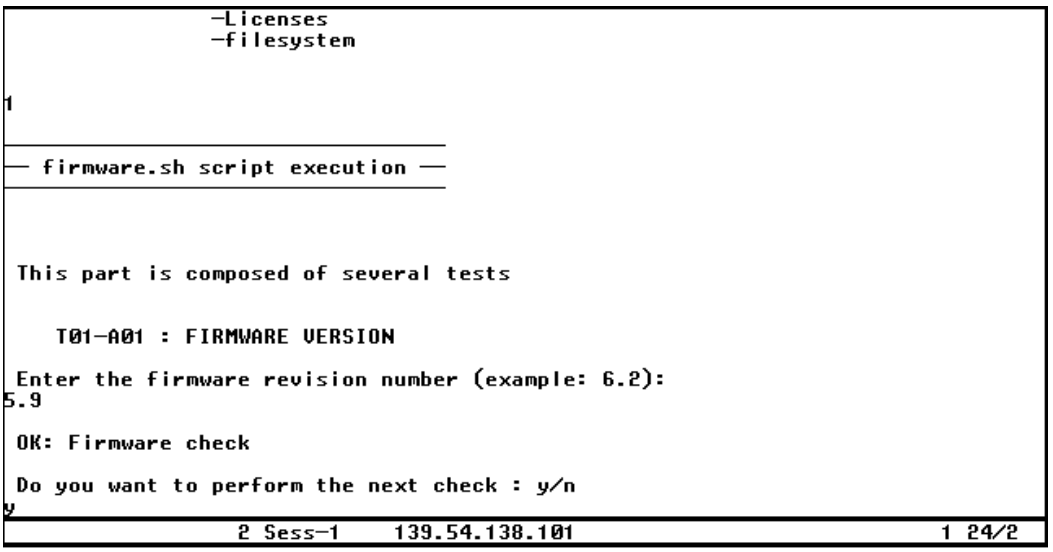

Figura 29: Elección de la opción 1

Vemos como el principio de cada script es indicado. Mientras que si elegimos cualquier otra opción que no sea ni la 1 ni la 0, se empieza ejecutando el primer test de la opción elegida:

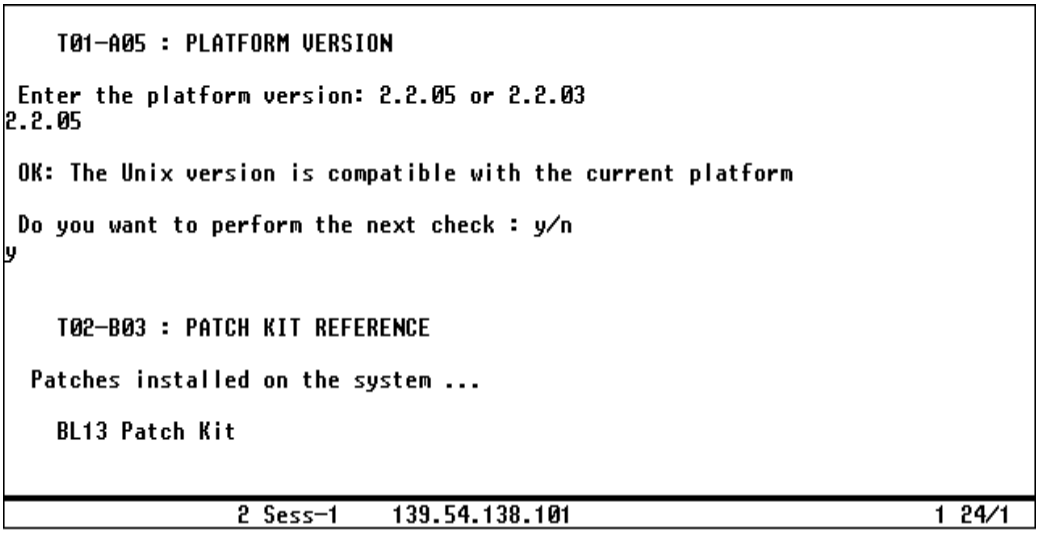

Figura 30: Ejecución de un test

En ambos casos, podemos observar que el número y nombre del test es indicado antes de ser ejecutado. A continuación, el parámetro necesario para la verificación es pedido e introducido seguidamente por el instalador y por último el resultado del test es indicado con un OK o NOK.

Entre un test y el siguiente de un mismo script una cuestión es introducida para preguntar la voluntad del operador de continuar o no con el siguiente test:

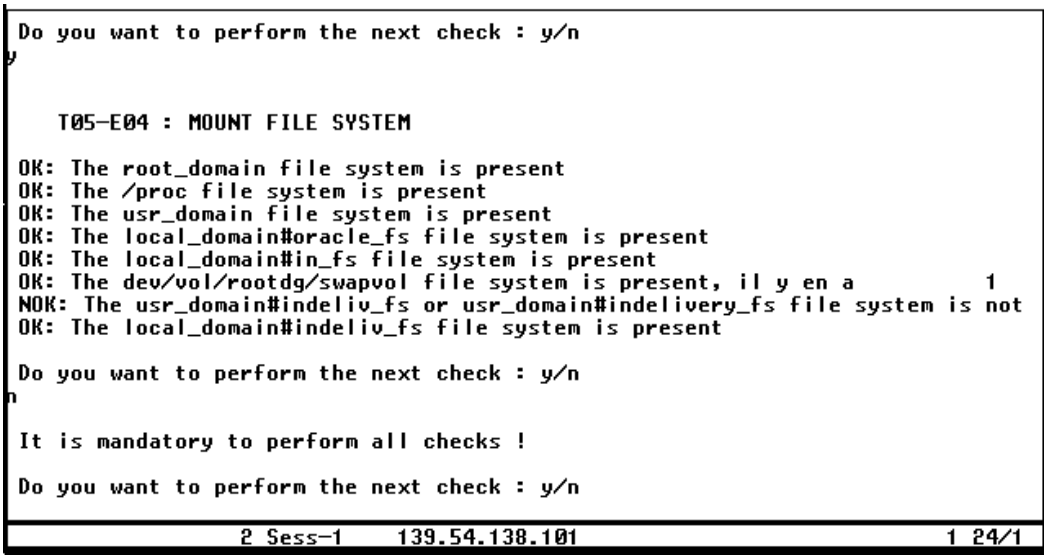

Figura 31: Cuestión entre tests

Como ejemplo de resultados no favorables expongo:

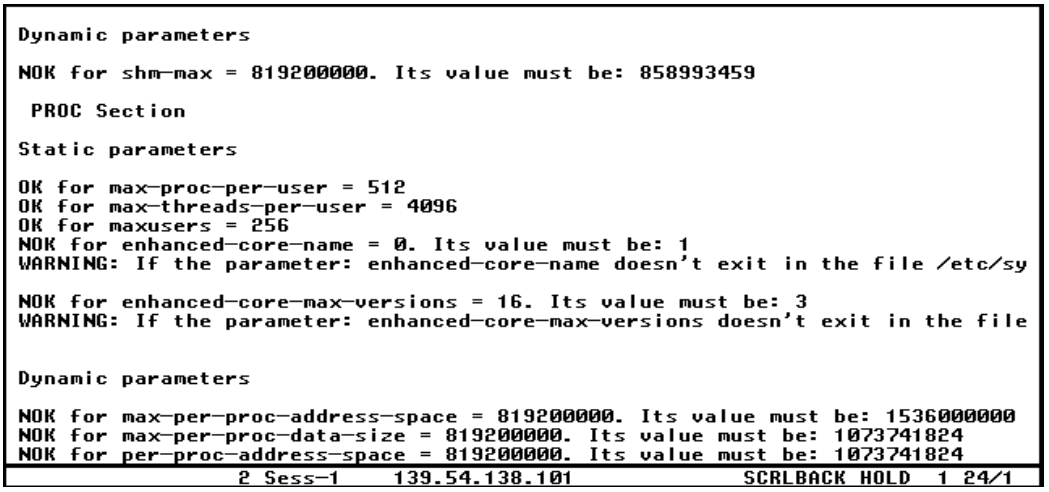

Figura 32: Resultados por pantalla

Todos estos resultados favorables o no, no sólo estarán indicados por pantalla sino también serán guardados en uno de los ficheros de resultados.

7. Una vez que el script ha finalizado, vuelve a la pantalla del menú principal, dándonos la posibilidad de elegir otra opción. Este tipo de ejecución de tests uno por uno, es muy útil en el caso de que hayamos ya lanzado una vez todos los tests seguidos y hayamos encontrados ciertos errores.

Cuando los corregimos, en lugar de pasar todos los tests otra vez, sólo paso el test o tests que me interesan, es decir, donde he encontrado el error de instalación.

En cualquiera de los casos y de las versiones, justo antes de salir del script principal, los ficheros de resultados de la verificación automática son recordados por pantalla.

4.2.3.4 Documentación y entrega

Una vez desarrollados los scripts de verificación, se me pidió escribir una documentación en la cual explicase los procedimientos descritos anteriormente. Sólo de esta forma la herramienta sería útil y utilizable por todos los interesados. Por ello, creé un documento que explica como realizar la verificación automática, paso a paso, y qué hacer con los resultados de dicha verificación.

Así mismo, también realicé un documento que se llama "readme.doc" para la gestión de versiones de scripts ya que éstos pueden ser modificados y mejorados posteriormente. En él expongo el número y nombre de scripts que hay para la verificación. También lo que automatiza cada uno de ellos. Es realmente un documento muy útil y es necesario leerlo antes de la propia verificación.

Una vez que tuve todo esto, realicé un paquete llamado Check_V1.zip que contenía:

- 21 scripts automáticos de control
- El documento de utilización de la herramienta
- "readme.doc" (detallado en el **Anexo B**)
- Los documentos automatizados
- Ejemplo general del fichero conf_file necesario sólo en el caso de que la instalación fuese manual.

A continuación hice la entrega del paquete de automatización, es decir, lo introduje en el servidor de mi servicio, el MAP. De esta forma, está a disposición de todos los ingenieros RTM del servicio.

4.2.3.5 Otros Tests y Optimización de los scripts

La llegada de la plataforma IN destinada a Afganistán (FULL IN monolítica duplex) fue la primera oportunidad para probar los scripts sobre una plataforma distinta a la del MAP. Es decir, una configuración diferente a la de "KIM".

Esta plataforma no sólo me permitió probar los scripts sino también comprobar la utilidad de los documentos redactados. Saber si el procedimiento de utilización era lo suficientemente claro para una persona que no conociese la herramienta. Por ello, pasé los scripts en compañía de un RTM que tenía como misión indicarme sus observaciones.

Esta verificación automática fue bastante satisfactoria y beneficiosa: aunque por un lado, encontré cosas que había que mejorar en los scripts, por otro lado, me permitió encontrar errores en la plataforma y corregirlos. El RTM me aconsejó, entre otros, sobre los textos de los mensajes de error para que éstos fueran más precisos.

Una vez finalizada la verificación automática, los ficheros de resultados son recuperados de la plataforma y puestos en el servidor MAP. De esta forma, tenemos un rastro del estado de la plataforma verificada antes de salir de Vélizy y cualquier persona interesada puede acceder a ellos.

A partir de esta verificación, comencé a mejorar los tests existentes e incluso añadí a estos otros nuevos gracias a la información que encontré por diferentes fuentes:

- **Los miembros del sevicio MAP**: me informaban sobre los documentos que recientemente habían sido puestos al día y los cuales tenía que tener en cuenta.

  Por ejemplo, los valores del fichero */etc/sysconfigtab* han sido modificados varias veces a lo largo de las prácticas. Significando un correspondiente cambio en el script **capa.sh**. Estos valores del Kernel son bastante importantes en la instalación de la plataforma y es inadmisible tener un error en este fichero.

- **El manager del equipo**: me comunicaba los tests de más que quería tener presentes en una verificación automática. Estas pruebas o tests no se encontraban en los documentos de verificación manual automatizados.

  Estos tests tenían por objetivo verificar cosas que eran bastante importantes en la instalación de una plataforma pero que en cambio antes no se comprobaban a mano. Es el caso de los tests para la configuración de "COCs" o de los comandos "ss7man", añadidos en el script **ss7.sh**.

- **Las verificaciones de las plataformas con scripts**: tuve la oportunidad de poder pasar los scripts por todos los tipos de configuraciones.

  Por entonces, había casi cada semana una plataforma que había que verificar. Esto me permitió ejecutar todos los tests y scripts en todos los tipos de máquinas: SMP, SMP/SCP, SCP, BEP y FEP.

  A veces, cuando encontraba algo que no marchaba muy bien en los scripts, podía cambiarlo inmediatamente y someterlo a un test sobre la misma plataforma. En caso contrario, debía esperar a acabar la verificación ya que debía reflexionar y buscar una solución.

- **Los instaladores en Vélizy y a distancia**: los instaladores una vez que conocieron la verificación automática, ya fuese por una demostración de mi parte, o por medio de un RTM responsable de las plataformas, empezaron a utilizarla y a comunicarme sus observaciones, las cuales fueron muy útiles.

Pude verificar durante esta etapa de mis prácticas al menos 7 plataformas. Al finalizar cada control, los ficheros de resultados fueron puestos sobre el servidor del MAP. Esto permitía llevar un control de la plataforma una vez entregada al cliente.

Estos tests me permitieron hacer una primera evaluación de la ganancia de tiempo aportada por los scripts. Así mismo, fueron útiles para formar a los instaladores y a los RTM y encontrar posibles mejoras de los programas.

La introducción de nuevos tests en los scripts me obligó a modificar el fichero de configuración aportándole nuevos datos. Ya que estos nuevos tests demandaban información que no estaba disponible en dicho fichero.

Además, comenzamos a enviar los scripts a los instaladores in situ que reenviaban los resultados al RTM responsable de la plataforma.

El procedimiento que seguí para la optimización de los scripts es descrito en el esquema siguiente:

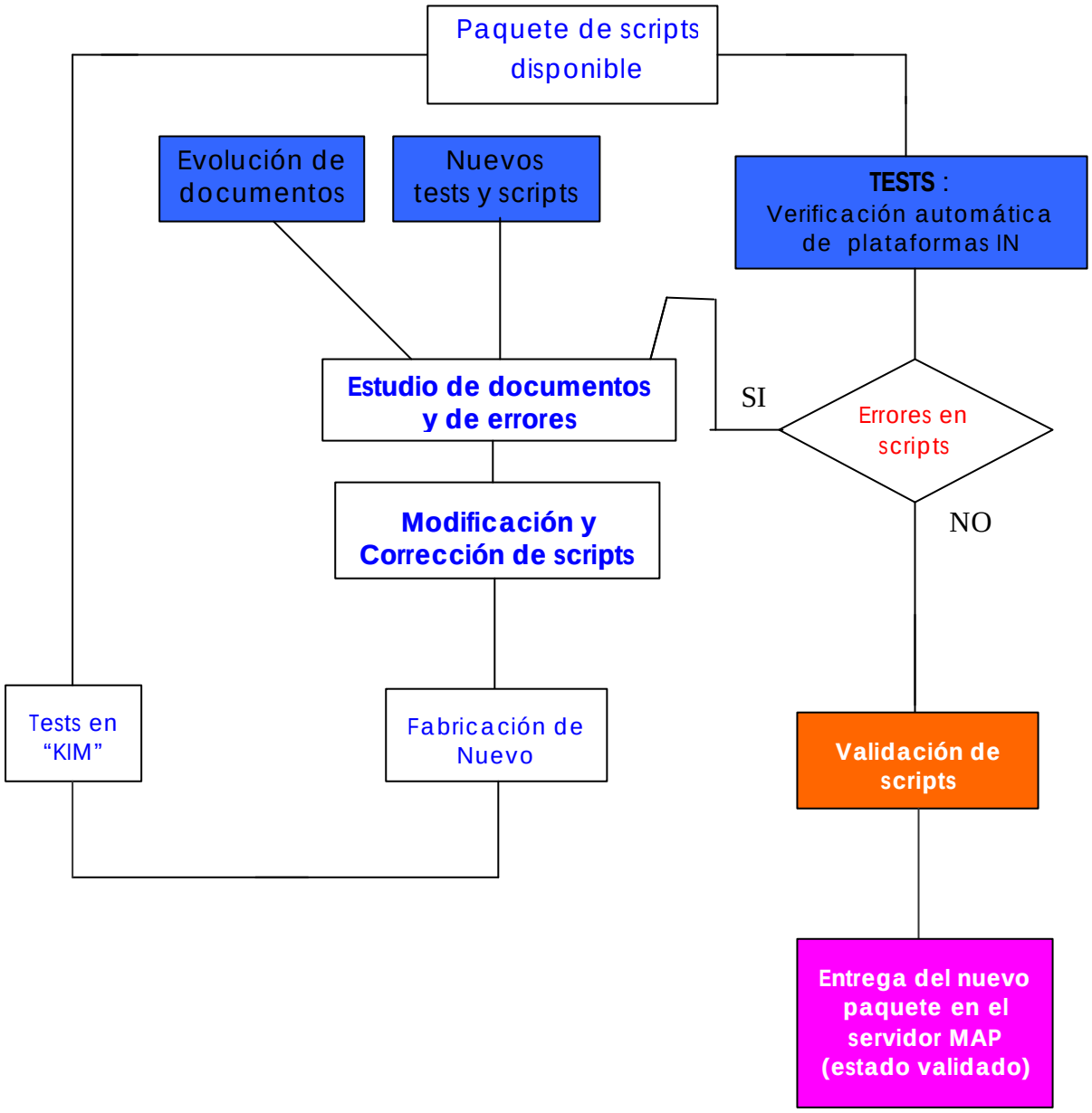

Figura 33: Proceso de optimización de scripts

Siguiendo este procedimiento entregué dos paquetes más con diferentes mejoras: primero Check_V2.zip y luego Check_V3.zip. La documentación correspondiente también fue puesta al día. En espera de otras plataformas, los probé y validé en primer lugar sobre "KIM", para asegurarme, al menos, que no había errores de sintaxis.

4.2.3.6 Últimos scripts y entregas

La instalación de los niveles llamados: 004, 027 y 115 es una de las partes más importantes de la instalación de una plataforma. Además, existe un "addendum" para cada nivel pero sólo se instala cada vez el "addendum" correspondiente al último nivel instalado en la plataforma. Por ejemplo, si se instala hasta el nivel 027, entonces instalamos el "addendum 027". Lo normal es instalar los tres niveles y el "addendum" del nivel 115.

Los scripts que existían, llamados **levelxxx.sh**, hacían el control del nivel instalado pero sin tomar en consideración el "addendum". Utilizan ficheros de extensión ".inv" (inventory) que contienen ciertos datos de una serie de ficheros de la plataforma. Estos ficheros son modificados o son creados durante la instalación de los niveles.

La verificación automática de nivel consistía en comparar con un software los datos de los ficheros de referencia, ficheros ".inv", y los ficheros existentes en el servidor después de la instalación.

La tarea que me fue encomendada era realizar un script destinado a la automatización de la verificación del nivel de "middleware" y del nivel de "addendum 115".

Escogimos el "addendum 115" porque es el más instalado. Y, conforme con el *principio 2 de automatización*, la verificación del "addendum 027" no debe ser incluida en los trabajos de automatización. Ya que no aportará una ganancia de tiempo realmente importante al no ser instalado casi nunca.

Existen a su vez diferentes niveles del "addendum 115". Y cada uno corresponde a un conjunto de parches.

La primera parte de esta tarea la dediqué a hacer ficheros de extensión *.inv* con todos los ficheros que eran modificados por el "addendum".

Para ello, tuve que analizar y estudiar el documento del "addendum 115". Aunque en este documento no había todos los ficheros que eran modificados, gracias a los miembros de mi equipo y a los resultados de otras verificaciones, conseguí hacer un fichero de extensión *.inv* para cada nivel del "addendum 115".

Estos ficheros se llaman: **nombreparch-adden115_niveladdendum.inv**.

También creé otro fichero general para todos los niveles llamado: **volatiles.inv** donde introduje todos los ficheros de la plataforma que son modificados manualmente, es decir, ficheros de contenido texto.

El script resultado se llama **level-adden115.sh**.

Vamos a ver ahora un ejemplo de la verificación automática del nivel y del "addendum 115". Es decir, como se presentaría de cara al operador la ejecución del script de verificación level-adden115.sh:

1. El primer paso es elegir la opción correspondiente a dicho script. Esta opción está presente en el menú principal del script **check_system.sh**:

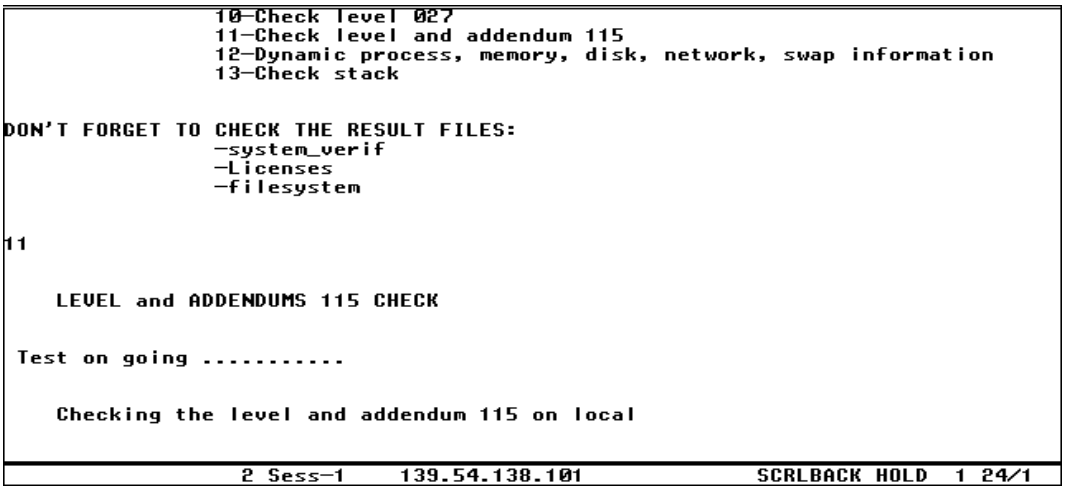

Figura 34: Verificación del nivel y addendum 115

2. Seguidamente empiezan a aperecer los resultados de los distintos tests que realiza:

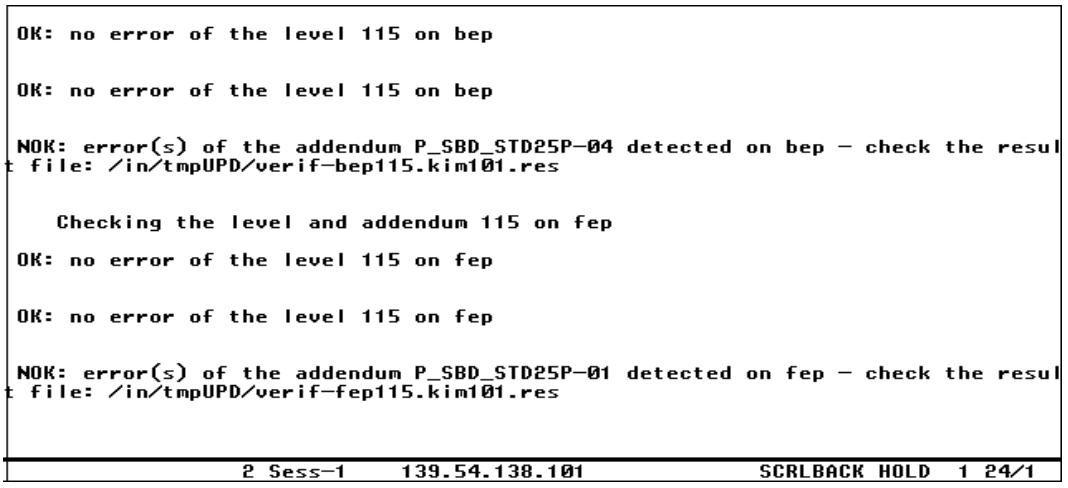

Figura 35: Resultados del script level-adden115.sh

Para acabar con la automatización de la verificación, realicé un script: **stack.sh** que verifica la correcta instalación de lo que llamamos "versión de stack SS7". La "stack SS7" no es mas que un conjunto de ficheros con los parámetros de las conexiones con la red de señalización. Gracias a los cuales podemos utilizar el protocolo Decnet.

Concretamente lo que automatiza dicho script es la verificación del documento: *DECss7 31J P10 Patch Kit for R2.2 IN platform*, documento de instalación de la "stack SS7". Este script significa una nueva opción en el menú del script principal de control del sistema.

Debido a estos últimos scripts y a sus correspondientes tests en las plataformas destinadas a Malawi y a Nigeria, entregué dos nuevos paquetes: Check_V4.zip y Check_V5.zip (el actual).

Todos los scripts se desarrollaron como previsto necesitando sólo algunos arreglos menores, principalmente de presentación de resultados.

Para una descripción detallada del código de los scripts remítase al **Anexo B**.

4.3 Automatización del proceso de instalación de las plataformas IN

En primer lugar, hay que distinguir dos partes muy diferentes en la instalación de una plataforma IN:

- Las partes del "Tome Volume" que sólo son verificaciones. No hay que confundirlas con las partes de control post-instalación automatizadas en el apartado anterior.

- Las partes del "Tome Volume" que implican modificaciones.

Así mismo, concerniendo a las fases de instalación que implican modificaciones, existen también dos partes dentro de la instalación automática:

- **Generación automática de ficheros sistema**, aplicación realizada en Visual Basic que genera los diferentes ficheros sistema de una plataforma IN.

- **Scripts de instalación**, el script que permite obtener los ficheros de configuración es incluido en este grupo aunque no forme parte directamente de la instalación.

4.3.1 Mejora y soporte de la aplicación: ServeurConfiguration.exe

Siendo la automatización de la instalación un proyecto que se extendía sobre una duración bastante importante, se trataba de garantizar la continuidad del trabajo de mi predecesor. Así pues, tenía que asegurar el correcto funcionamiento y el apoyo de la herramienta de instalación realizada antes de mi llegada.

4.3.1.1 Descripción de la herramienta

La aplicación en Visual Basic realizada por el anterior becario en el servicio genera los ficheros sistema que requieren más modificaciones durante la instalación de la plataforma.

Este programa necesita que se le proporcione como entrada el tipo de configuración de la plataforma (STEP IN, FULL IN monolítica o distribuida), los hostnames (o nombres que el cliente proporciona) de los servidores, los nombres lógicos de los BEPs y las direcciones IP de cada elemento conectado a cada LAN (servidores, alias, PC Remoto y opcionalmente gateway y PC de explotación).

Como salida, genera los ficheros siguientes (precedidos del camino de acceso al directorio al cual pertenecen en los servidores):

- */etc/hosts* y */etc/routes* para el rutado.

- */in/local/conf/db/listener.ora* que contiene la declaración de todos los servicios instalados.

- */in/local/etc/hostinfo.ini* para configurar el proceso "Netwatcher".
- */in/local/conf/smpmon.cfg* para la configuración de "Alarm Box".
- */in/smp/lms/sys/cfg/host.cfg, /in/smp/lms/sys/cfg/cnx.cfg* y */in/smp/lms/sys/cfg/sys.cfg* para el LSM que administra el balanceo o cambio del dominio compartido de un SMP al otro.
- */in/smp/web/etc/apache/httpd.conf* para el servidor "Apache" con el fin de tener acceso, usando un navegador web, a las herramientas "Accman" y "Comprov" para la gestión de la plataforma.

La generación de los ficheros precede a la instalación de la plataforma. De esta forma, los ficheros creados están disponibles al principio de ésta.

Estos ficheros serán transferidos a los servidores pasando por el PC Remoto. Los ficheros son trasladados bajo nombres diferentes a los verdaderos con el fin de verificar su contenido antes de reemplazarlos. Después de ser verificados, serán copiados en los servidores en determinados momentos de la instalación (indicados en el "Tome Volume") y así sustituirán a los antiguos ficheros.

4.3.1.2 Problemas a resolver

A raíz de varias utilizaciones de la aplicación *ServeurConfiguration.exe*, encontramos una limitación del programa y algunos pequeños errores en los contenidos de los ficheros generados automáticamente:

- Era imposible asignar una dirección al RPC sobre la LAN1 (limitación).
- Faltaban ciertos nombres lógicos en el fichero */etc/hosts.*
- Línea desplazada en el fichero */in/smp/web/etc/apache/httpd.conf*.
- Función incorrecta de los servidores BEPs y FEPs en el fichero */in/local/etc/hostinfo.ini*.
- Error en el fichero */in/local/conf/db/listener.ora*.

4.3.1.3 Solución: script de corrección

Después de haber analizado los problemas, tenía dos soluciones posibles:

- modificar directamente las rutinas de la aplicación en Visual Basic,
- hacer un script para corregir los ficheros generados con el programa.

Hay que tener en cuenta que la automatización no puede implicar un coste en tiempo de programación muy superior con relación a la ganancia aportada. Así pues, decidí con el acuerdo de mi tutor de prácticas, que la mejor opción era la de realizar un script de corrección (según el *principio 2 de automatización*).

El script resultado se llama **modif.sh** y es ejecutado sobre el PC Remoto, antes de que los ficheros sean transferidos a los servidores. Lo probé y validé sobre una plataforma FULL IN monolítica.

A continuación modifiqué la documentación de utilización de la herramienta y puse sobre el servidor MAP la nueva entrega del paquete del programa *ServeurConfiguration.exe* que contiene todo lo necesario para la correcta generación de los ficheros sistema.

4.3.2 Scripts automáticos de instalación

4.3.2.1 Definición de las necesidades

Según las demandas de los instaladores, los procesos que hay que automatizar son los que implican la escritura de largas cadenas de caracteres, tanto en los comandos como en las modificaciones de ficheros, así como todos los procesos muy repetitivos.

Los RTMs, por su parte, recomiendan una automatización completa de las fases o partes de comprobación del "Tome Volume".

Los objetivos de la instalación automática son:

- ganar tiempo sobre la instalación manual y
- aumentar su fiabilidad (menos errores humanos).

4.3.2.2 Principios y obligaciones

La automatización de la instalación debe respetar tanto los *principios de automatización* (ver epígrafe 3.1.2) como las especificaciones siguientes:

- Reutilización de la herramienta existente: los ficheros sistema generados antes de la instalación son una fuente importante de informaciones.
- Tener en cuenta todas las posibles configuraciones de plataforma IN Alcatel R2.2.05: los scripts deben ser aplicables a cualquier tipo de plataforma que haya que instalar.
- Seguridad mínima: conservar las antiguas versiones de los ficheros antes de ser modificados por los scripts.
- El Shell utilizado será el Shell de Bourne para garantizar una compatibilidad máxima con todos los sistemas UNIX.
- La lengua utilizada será el inglés ya que así muchos instaladores extranjeros podrán utilizar estos scripts.

Es necesario tener una gran prudencia con todo lo que implica modificaciones en el sistema o en las aplicaciones que funcionan sobre la plataforma, un dato erróneo puede provocar un funcionamiento incorrecto grave o, peor aún, poner fuera de servicio la plataforma (conforme con el *principio 3* de automatización).

Además, ciertas partes de la instalación no se pueden automatizar debido a la necesidad de intervención física por parte del instalador sobre la plataforma.

4.3.2.3 Utilización de scripts

Una vez analizadas las partes que se podían automatizar según los principios de automatización y recomendaciones, me dispuse a desarrollar los scripts.

El procedimiento que seguí para ello fue similar al explicado para los de verificación, sólo que en este caso no podía probarlos sobre "KIM" siempre que quisiera por razones que explicaré en el siguiente apartado. Únicamente las partes de verificación eran probadas cada vez.

Creé un directorio llamado "log" (*/in/tmp/check/log*) que contenía todos los ficheros de registro con todos las instrucciones realizadas por los scripts y todos los resultados obtenidos. Éstos eran puestos al día durante la ejecución de los scripts de instalación.

Así mismo, creé un directorio llamado "bak" (*/in/tmp/check/bak*) donde introducía todas las versiones originales de los ficheros sistema antes de ser remplazados automáticamente. En caso de error, siempre podría recurrir a dichos ficheros.

Esta organización de ficheros facilita la consulta de las acciones realizadas por los scripts y la recuperación de los antiguos ficheros sistema.

Los parámetros necesarios son igualmente buscados directamente en el fichero de configuración específico al servidor.

El método de ejecución elegido para la instalación automática es el siguiente:

1. Generación de los ficheros sistema con la aplicación *ServeurConfiguration.exe.*

2. Almacenamiento de los ficheros y los scripts en el RPC.

3. Ejecución del script encargado de generar los ficheros de configuración: **T00_conffile_generator.sh**. Como en el caso de la verificación automática, los scripts necesitan ciertas informaciones en el transcurso de su funcionamiento. Y éstos les serán facilitados también por medio del fichero de configuración.

4. Copia de los ficheros de configuración, distinguidos por el nombre del servidor al cual cada uno está destinado, en sus servidores respectivos.
   Copia de los ficheros sistema y de los scripts de instalación en cada servidor.

5. Lanzar el script **menú.sh** que contiene un menú cuyas opciones son las distintas partes de instalación que se pueden realizar automáticamente. Escoger la parte que se quiere realizar en cada nueva utilización de los scripts.

Existe un documento para cada parte del "Tome Volume" automatizada que aclara la utilización de los scripts. Este documento es hecho a partir del "Tome Volume" y muestra la correspondencia entre la parte automatizada y el script que la realiza.

6. Borrar el directorio de almacenamiento, */in/tmp/check*, una vez acabada la instalación. Esto permite no dejar en los servidores ningún fichero no deseado por el cliente.

Y el esquema funcional que describe este procedimiento de utilización es:

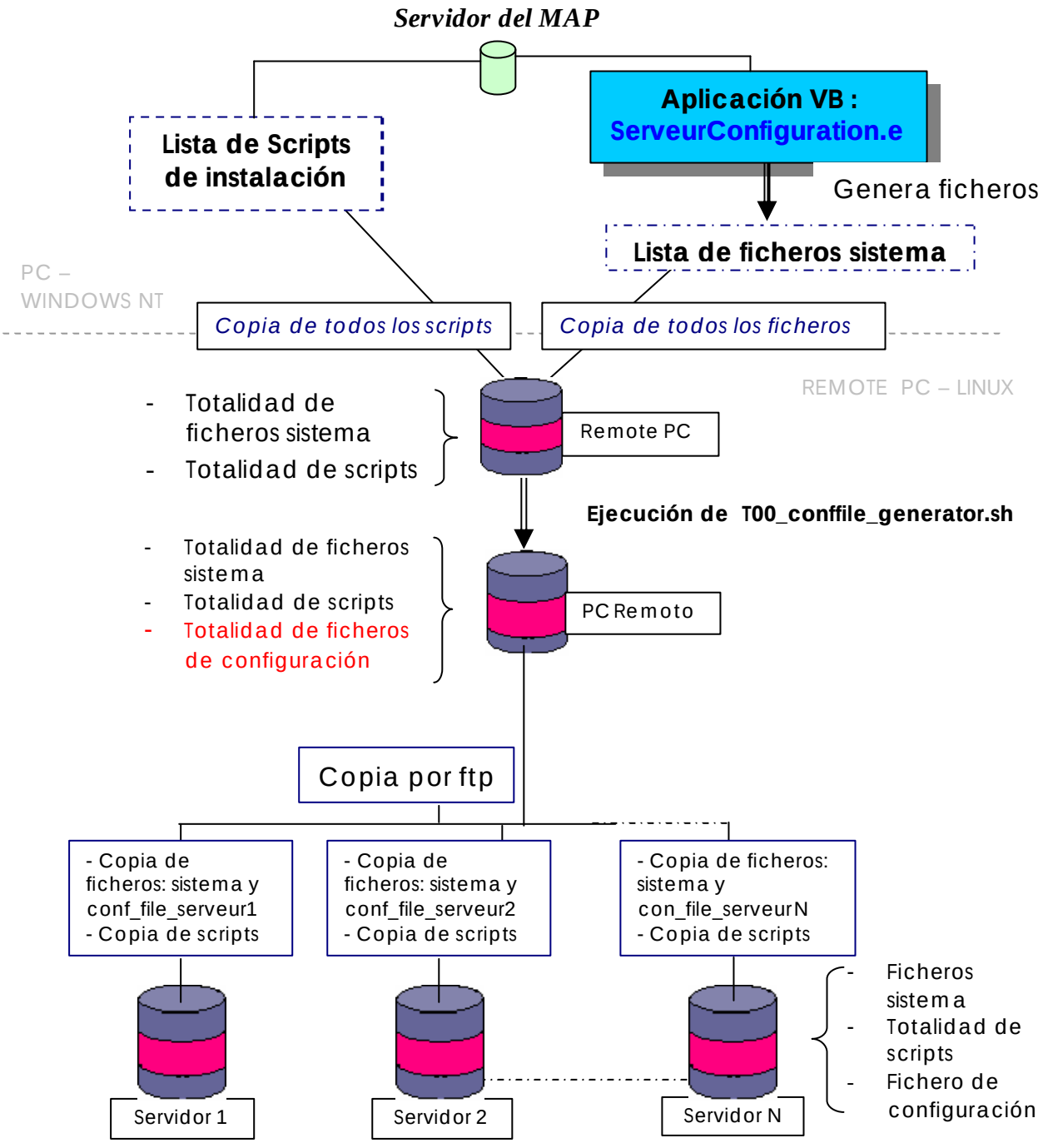

Figura 36: Procedimiento de utilización de los scripts de instalación

El almacenamiento de los ficheros sistema y de configuración generados y los scripts se realiza en el directorio de trabajo del servidor llamado */in/tmp/check*.

4.3.2.4 Tests y validaciones

En primer lugar, hay que aclarar que si bien para el caso de los scripts de verificación, podía disponer prácticamente de todos las plataformas IN que pasaban por Vélizy para probar, validar y mejorar los programas, en el caso de la instalación no era así. Mas bien ocurría todo lo contrario.

El problema era que para probar los scripts de instalación necesitaba una plataforma sin instalar, es decir, una plataforma que viniese directamente de la fabrica de HP o que hubiera sido desinstalada. Con lo cual, la plataforma del MAP, "KIM", no me servía o al menos en parte ya que no podía probar los scripts que automatizaban las fases que modificaban la misma.

Además, estos programas de instalación automática necesitan mucha mas precaución a la hora de ejecutarlos y por lo tanto mucho mas tiempo para probarlos.

Por todo ello, sólo cuando una plataforma era recuperada del cliente al ser sustituida por otra o bien cuando no teníamos excesiva prisa por entregarla al cliente después de Vélizy, podía utilizarla. Y esto rara vez ocurría porque como en toda cadena de producción, el producto debía salir lo más rápido posible hacia el cliente.

Una vez aclarado esto, continúo ahora con el primer test de los scripts de instalación.

La primera plataforma sobre la cual pude probar estos programas estaba compuesta únicamente por dos SMPs. Esto limitó las pruebas a las funcionalidades concernientes al SMP.

Lo ideal hubiera sido someter a test todos los scripts sobre todos los tipos de servidores existentes ya que dichos scripts son diferentes según el tipo de servidor.

Este primer test lo realicé siguiendo el procedimiento descrito en el apartado anterior y en compañía de un instalador que se encargaba de las partes manuales de la instalación.

Aunque obtuve resultados muy positivos de este primer test (validé algunos scripts), también encontré algunas cosas que necesitaban ser mejoradas. Incluso tuve que decidir, con ayuda de mis compañeros, la supresión de una parte de uno de los scripts ya que no era factible automáticamente y originaba graves errores.

Después de esta prueba, seguí trabajando sobre estos scripts, mejorando algunos de ellos y poniendo al día otros. A veces era posible corregir partes destinadas a servidores que no fueran SMPs ya que ciertos comandos eran similares.

El script que genera los ficheros de configuración, **T00_conffile_generator.sh**, fue también modificado para tomar en cuenta los nuevos parámetros introducidos en el fichero.

En espera de la llegada de nuevas plataformas aptas para mis pruebas, lancé sobre "KIM" únicamente las partes de verificación. Esta segunda prueba celebrada sobre la plataforma del MAP me permitió validar completamente los scripts que automatizan las partes de comprobación del "Tome Volume".

Por aquel entonces se pensó que, gracias a la clonación de “KIM”, se podría desinstalar, es decir, volverla a la capa HP y realizar una nueva instalación completa de esta plataforma con los scripts. Pero debido a problemas de origen hardware, no pudimos devolverla a tiempo al nivel de la capa HP (nivel de instalación en la plataforma cuando llega de la fábrica de HP).

En consecuencia, las partes de comprobación de parámetros, de ficheros y de conectividad pudieron ser validadas con precisión. Pero las partes que modifican el sistema necesitan ensayos o tests en situación reales (una primera instalación de la plataforma).

Creé así un cuadro de pruebas que contiene toda la información sobre el estado de validación de los scripts en función del tipo de servidor y de la configuración. El estado de un script para un cierto servidor puede ser:

- Probado y validado,
- Probado y validado sobre otro tipo de servidor pero debe también funcionar sin riesgo para el servidor considerado,
- A validar incluso si ya ha sido sometido a un test sobre otro tipo de servidor o configuración,
- Inútil para el tipo de servidor considerado porque no tiene que ser realizado para el mismo.

De esta forma, cualquier persona que desee utilizar estos scripts, podrá saber de antemano en que situación se encuentran. Para mas información sobre el formato de este cuadro de tests, remítase al **Anexo C**.

4.3.2.5 Resultados de los scripts de instalación

La gran mayoría de las partes del “Tome Volume” automatizadas son de comprobación.

Además ciertas partes de los documentos que habían sido escogidos en un principio resultaron ser no factibles automáticamente, como por ejemplo el reinicio de los servidores. Así pues, los documentos de utilización de los scripts indican claramente que partes o comandos son hechos por ellos y que script corresponde a cada parte.

El cuadro (o tabla) siguiente proporciona el porcentaje del documento o de la operación que ha sido finalmente automatizado:

%	Documentos u Operaciones
70	Verification and update of the date and the hour
60	IP address, hostname and DECnet configuration of Compaq platform
90	Network verification of Compaq platform
100	Hardware Verification of Compaq platform
50	SS7 Interfaces Verification of Compaq platform
60	middleware Installation (PFD_STD25M)
100	Software licence patch verification of Compaq platform
100	Unix Kernel verification of Compaq platform
100	Disks verification of Compaq platform

Figura 37: Cuadro de etapas del Tome Volume automatizadas y sus porcentajes

4.3.2.6 Ejemplo

Como hemos visto anteriormente una de las operaciones automatizadas es la comprobación y el ajuste del reloj de los servidores de la plataforma. El script que la realiza es: **T02B04_Verification_and_Update_of_the_Date_and_the_Hour.sh**.

Para la utilización del mismo procedemos como sigue:

1. Se ejecuta el script **menu.sh** y obtenemos por pantalla el menú principal:

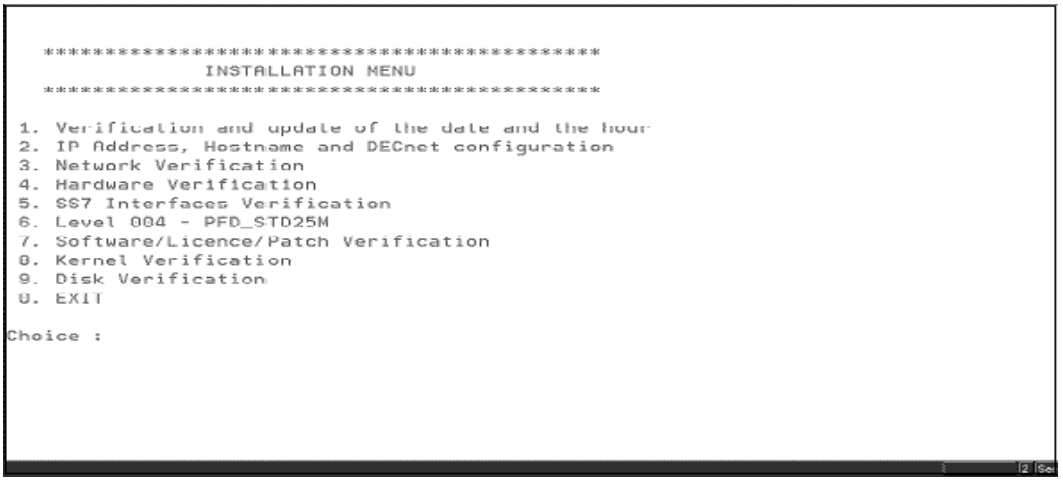

Figura 38: Menú principal de la instalación

2. Elegimos la opción 1 y la ejecución del script da como resultado:

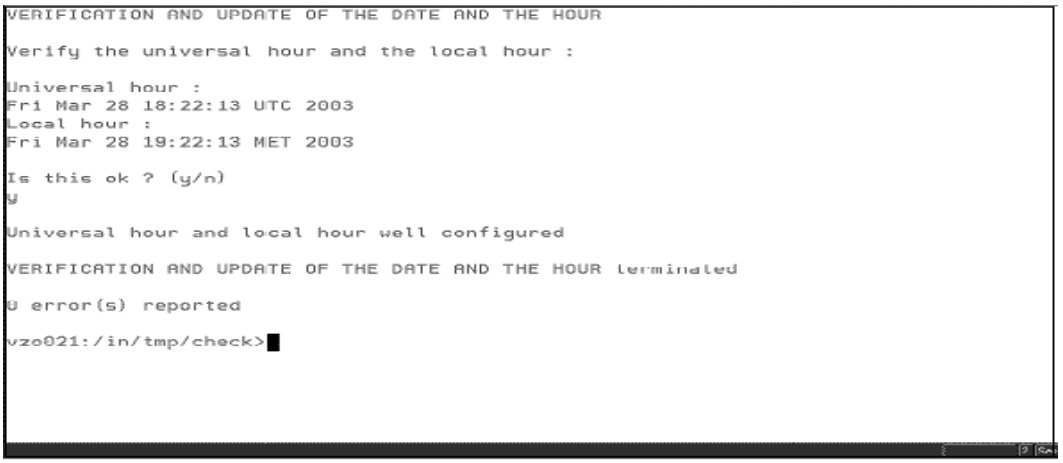

Figura 39: Resultados de la comprobación de la hora

En este caso concreto, vemos que la hora universal y la zona horaria son correctas. Para asegurarse, el programa nos pregunta si realmente estos datos son correctos y al responder "sí" termina su ejecución indicándonos antes el número de errores que se han encontrado a lo largo de su desarrollo.

Si por el contrario no estuviéramos de acuerdo con la hora o la zona, el script nos da la oportunidad de cambiarla e introducir la correcta indicándonos los pasos a seguir.

3. El fichero de extensión ".log" correspondiente registró precisamente todo lo que se indicó por pantalla durante la ejecución del script:

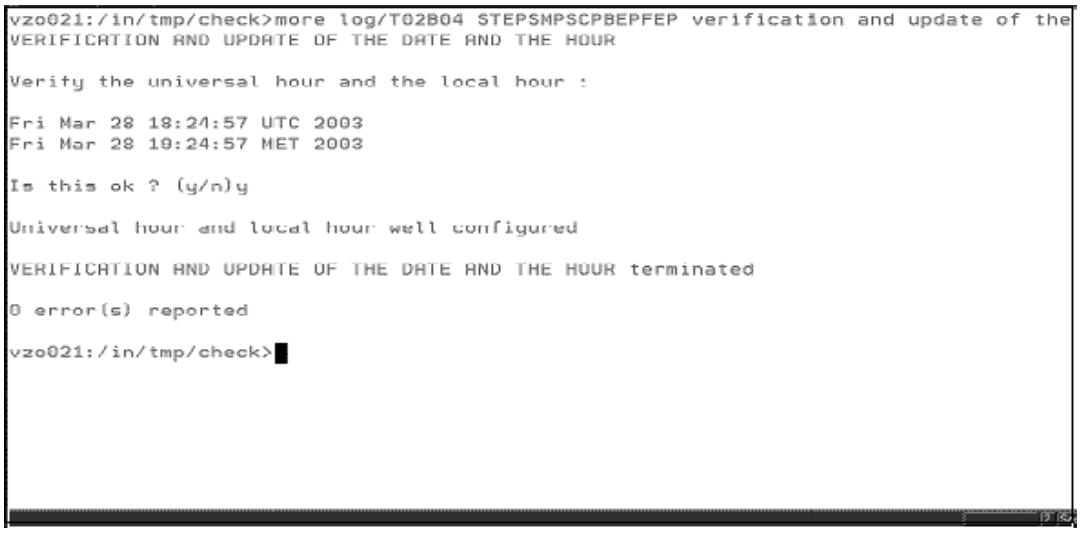

Figura 40: Contenido de un fichero de registro