



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

Índice Tema 8

1. Los lenguajes de control de trabajos: concepto, estructura, elementos y funcionalidades.
 - 1.1. Concepto.
 - 1.2. Estructura, elementos y funcionalidades.
2. Las técnicas y herramientas de operación automática.



CENTRO DE ESTUDIOS FINANCIEROS

VIRIATO, 52	28010 MADRID	914 44 49 20
PONZANO, 15	28010 MADRID	914 44 49 20
G. DE GRÀCIA, 171	08012 BARCELONA	934 15 09 88
ALBORAYA, 23	46010 VALENCIA	963 61 41 99

www.cef.es

info@cef.es

TEMA 8

Los lenguajes de control de trabajos: concepto, estructura, elementos y funcionalidades. Las técnicas y herramientas de operación automática.

1. LOS LENGUAJES DE CONTROL DE TRABAJOS: CONCEPTO, ESTRUCTURA, ELEMENTOS Y FUNCIONALIDADES.

1.1. CONCEPTO.

En los primeros ordenadores de finales de los 40 hasta mediados de los 50, el programador interactuaba directamente con el hardware; no había sistema operativo. La operación con la máquina se efectuaba desde una consola consistente en unos indicadores luminosos, unos conmutadores, algún tipo de dispositivo de entrada y una impresora. Los programas en código máquina se cargaban a través del dispositivo de entrada (un lector de tarjetas, por ejemplo). Si se detenía el programa por un error, la condición de error se indicaba mediante los indicadores luminosos. El programador podía examinar los registros y la memoria principal para determinar la causa del error. Si el programa continuaba hasta su culminación normal, la salida aparecía en la impresora.

Estos primeros sistemas presentaban dos problemas principales:

- **Planificación:** la mayoría de las instalaciones empleaban un formulario de reserva de tiempo de máquina. Normalmente un usuario podía reservar bloques de tiempo en múltiplos de media hora o algo por el estilo. Un usuario podía reservar una hora y terminar a los 45 minutos, esto daba como resultado un desperdicio del tiempo del ordenador. Por el contrario, el usuario podía tener dificultades, no terminar en el tiempo asignado y verse forzado a parar sin haber resuelto el problema.
- **Tiempo de preparación:** un programa sencillo, llamado trabajo, cargaba un compilador y un programa en lenguaje de alto nivel (programa fuente) en la memoria, salvaba el programa compilado (programa objeto) y luego montaba y cargaba el programa objeto junto con las funciones comunes. Cada uno de estos pasos podía implicar montar y desmontar cintas o preparar paquetes de tarjetas. Si se producía un error, el infortunado usuario tenía que volver al inicio de este proceso de preparación. De este modo, se perdía un tiempo considerable en preparar un programa para su ejecución.

Este modo de operación podría denominarse proceso en serie porque refleja el hecho de que los usuarios tenían que acceder al ordenador en serie. Con el paso del tiempo se desarrollaron varias herramientas de software de sistemas para intentar hacer más eficiente este proceso en serie. Entre éstas se incluían bibliotecas de funciones comunes, montadores, cargadores y rutinas de manejo de E/S que estaban disponibles como un software común para todos los usuarios.

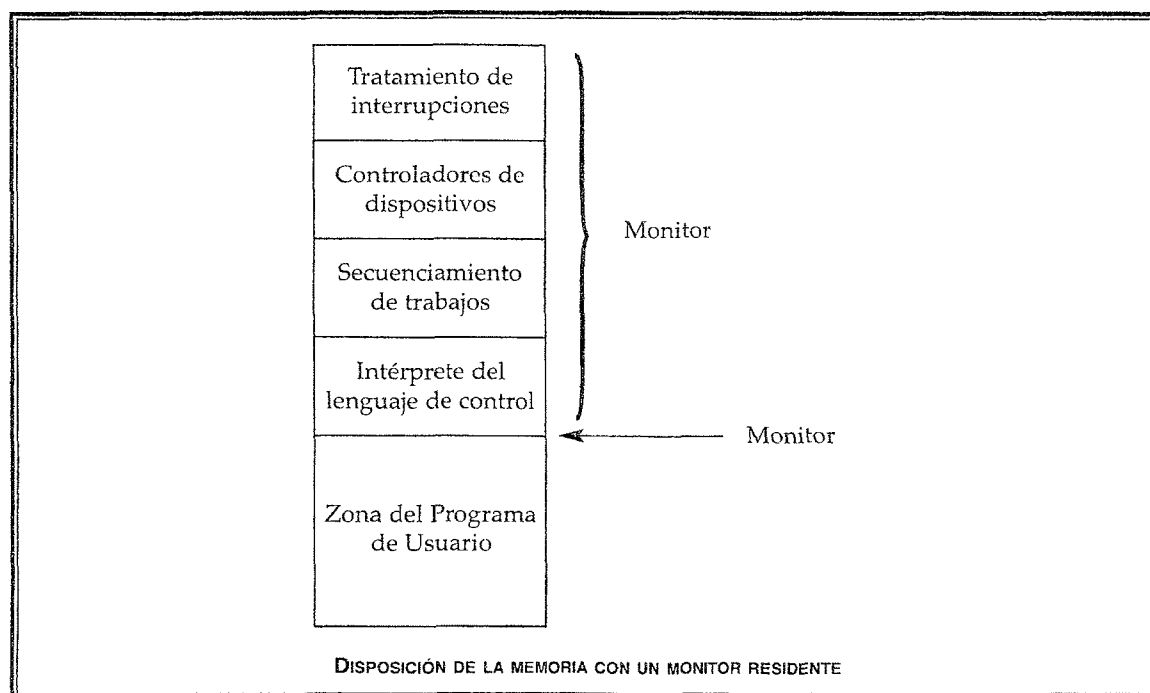
Las primeras máquinas eran muy caras y, por tanto, era importante maximizar la utilización de las mismas. El tiempo desperdiciado por la planificación y la preparación se mejoró con el concepto de sistema operativo por lotes (batch). A principios de los 60, un conjunto de constructores ya habían desarrollado sistemas operativos por lotes para sus computadores.

La idea central que está detrás del esquema sencillo de proceso por lotes es el uso de un elemento de software conocido como monitor. Con el uso de esta clase de sistema operativo, los usuarios ya no tendrán acceso directo a la máquina. En su lugar, el usuario deberá entregar los trabajos en tarjetas o en cinta al operador del ordenador, que agrupaba secuencialmente los trabajos por lotes y ubicaba los lotes enteros en un dispositivo de entrada para su empleo por parte del monitor. Cada programa se construirá de modo tal que volviera al monitor al terminar su proceso y, en ese momento, el monitor comenzaba a cargar automáticamente el siguiente programa.

Para entender cómo funciona este esquema, se va a ver desde dos puntos de vista: el del monitor y el del procesador. Desde el punto de vista del monitor, él es el que controla la secuencia de sucesos.

Para que esto sea posible, gran parte del monitor debe estar en memoria principal y disponible para su ejecución. Esta parte del monitor se conoce como monitor residente. El resto del monitor consta de utilidades y funciones comunes que se cargan como subrutinas en los programas de los usuarios al comienzo de cualquier trabajo que las necesite. El monitor lee los trabajos uno a uno del dispositivo de entrada (normalmente, un lector de tarjetas o una unidad de cinta magnética).

A medida que lo lee, el trabajo actual se ubica en la zona del programa de usuario y el control pasa al trabajo. Cuando el trabajo termina, se devuelve el control al monitor, que lee inmediatamente un nuevo trabajo. Los resultados del trabajo se imprimen y se entregan al usuario.



Considere esta secuencia desde el punto de vista del procesador. En un cierto momento, el procesador estará ejecutando instrucciones de la zona de memoria principal que contiene al monitor. Estas instrucciones hacen que el trabajo siguiente sea leído en otra zona de la memoria principal. Una vez que el trabajo se ha leído, el procesador encuentra en el monitor una instrucción de desvío que ordena al procesador continuar la ejecución en el inicio del programa de usuario. El procesador ejecuta entonces las instrucciones del programa de usuario hasta que encuentre una condición de finalización de error. Cualquiera de estos dos sucesos provoca que el procesador vaya por la instrucción siguiente del programa monitor. De este modo, la frase «el control se le pasa al trabajo» quiere decir simplemente que el procesador pasa a leer y a ejecutar instrucciones del programa de usuario, que la frase «el control vuelve al monitor» quiere decir que el procesador pasa ahora a leer y ejecutar las instrucciones del programa monitor.

Debe quedar clara que es el monitor el que gestiona el problema de la planificación. Se pone en cola un lote de trabajos y éstos son ejecutados tan rápido como es posible, sin que haya tiempo alguno de desocupación.

¿Qué ocurre con la preparación de los trabajos? El monitor también se encarga de esto. Con cada trabajo, se incluyen instrucciones de una forma primitiva de lenguaje control de trabajos (JCL, Job Control Language), que es un tipo especial de lenguaje de programación empleado para dar instrucciones al monitor. Veamos un ejemplo sencillo con entradas de trabajos desde tarjetas.

Para ejecutar el trabajo, el monitor lee la tarjeta identificativa del tipo de compilador y carga el compilador adecuado desde el dispositivo de almacenamiento masivo (generalmente una cinta). El compilador traduce el programa de usuario en código objeto, que se almacena en memoria en el dispositivo de almacenamiento. Si se carga en memoria, la operación es conocida como «compilar, cargar y arrancar» (compile, load and go). Si se almacena en cinta, entonces se requiere la tarjeta &LOAD. Esta tarjeta es leída por el monitor, que retoma el control después de la operación de compilación. El monitor llama al cargador, que carga el programa objeto en memoria en lugar del compilador y le transfiere el control. De esta manera, un segmento grande de memoria se puede compartir entre diferentes subsistemas, aunque en cada momento sólo uno de ellos tiene que estar presente y ejecutándose.

Durante la ejecución del programa de usuario, cada instrucción de entrada origina la lectura de una tarjeta de datos. La instrucción de entrada en el programa de usuario hace que se invoque una rutina de entrada, que forma parte del sistema operativo. La rutina de entrada se asegura de que el programa de usuario no ha leído accidentalmente una tarjeta JCL. Si esto sucede, se produce un error y el control se transfiere al monitor. Al terminar un trabajo, con o sin éxito, el monitor recorre las tarjetas de entrada hasta encontrar la próxima tarjeta JCL. De este modo, el sistema se protege contra un programa que tenga tarjetas de datos de más o de menos.

Se comprobará que el monitor o el sistema de proceso por lotes es simplemente un programa de computadora. Se basa en la capacidad del procesador para traer y ejecutar instrucciones desde varias zonas de la memoria principal y así apoderarse y ceder el control de forma alterna. Para esto serían convenientes algunas otras características del hardware, entre las que se encuentran las siguientes:

- Protección de memoria: mientras el programa de usuario esté ejecutándose, no debe modificar la zona de memoria en la que está el monitor. Si se hace un intento tal, el hardware del procesador deberá detectar el error y transferir el control al monitor. El monitor abortará entonces el trabajo, imprimirá el mensaje de error y cargará el siguiente trabajo.

- **Temporizador:** se utiliza un temporizador para impedir que un solo trabajo monopolice el sistema. El temporizador se lanza al comenzar cada trabajo. Si expira el tiempo, se producirá una interrupción y el control volverá al monitor.
- **Instrucciones Privilegiadas:** ciertas instrucciones son consideradas como privilegiadas y pueden ser ejecutadas sólo por el monitor. Si el procesador encuentra una instrucción tal, cuando está ejecutando el programa del usuario, se producirá una interrupción de error. Entre las instrucciones privilegiadas se encuentran las instrucciones de E/S, de forma que el monitor retenga el control de todos los dispositivos de E/S. Esto impide, por ejemplo, que un programa de usuario lea accidentalmente instrucciones de control que son del trabajo siguiente. Si un programa de usuario desea realizar una E/S, debe solicitar al monitor que haga la operación por él. Si el procesador encuentra una instrucción privilegiada cuando está ejecutando un programa de usuario, el hardware del procesador la considera como un error y transfiere el control al monitor.
- **Interrupciones:** los primeros modelos de computadores no tenían esta capacidad. Esta característica aporta al sistema operativo más flexibilidad para ceder y retomar el control de los programas usuarios.

Naturalmente se puede construir un sistema operativo sin esta característica, pero los fabricantes de computadores comprobaron que los resultados eran caóticos y, por tanto, incluso los sistemas operativos por lotes más primitivos ya disponían de estas características en el hardware. Por otro lado, hay que decir que el sistema operativo más utilizado del mundo, el MS-DOS, no dispone de protección de memoria ni de instrucciones privilegiadas de E/S. Sin embargo, como este sistema está destinado a computadoras personales de un solo usuario, los problemas que se puedan originar son menos graves.

En un sistema operativo por lotes, el tiempo de máquina se reparte entre la ejecución de programas de usuarios y la ejecución del monitor. Así se tienen dos pérdidas: se entrega al monitor cierta cantidad de memoria principal y éste consume cierto tiempo de la máquina. Ambas pérdidas son una forma de sobrecarga. Aun con esta sobrecarga, los sistemas operativos sencillos por lotes mejoran el uso de la computadora.

1.2. ESTRUCTURA, ELEMENTOS Y FUNCIONALIDADES.

¿Qué es JCL? Es el acrónimo de Job Control Language (Lenguaje de Control de Trabajos), es un medio de comunicarse con el IBM 3090 MVS Operating System. Las instrucciones JCL proveen de información para que el sistema operativo pueda ejecutar un trabajo (Job).

Vamos a ver las instrucciones básicas de JCL:

- Job.
- Execute.
- Data Definition.
- Comment.
- Delimiter.
- Null.

Job. Define el trabajo a realizar, puede constar de varios pasos conocidos como tareas (JobSteps). Su formato es el indicado a continuación:

```
//jobname JOB (*),username,TIME=(n,nn),NOTIFY=*,  
// CLASS=a,MSGLEVEL=(n,n),MSGCLASS=<I>a,TYPRUN=aaaa
```

A continuación se explican los diversos parámetros que lo componen:

1. jobname: nombre del trabajo (job) hasta 8 caracteres.
2. Username: nombre de la persona a la que pertenece el job.
3. CLASS: agrupa job's de acuerdo requerimientos de tiempo, cinta o impresora.
4. TIME: especifica el límite de proceso de tiempo en minutos y segundos.
5. NOTIFY: especifica a quién se notifica por parte del sistema.
6. MSGLEVEL: controla los mensajes de impresión y de asignación de recursos.
7. MSGCLASS: especifica el formato de salida de los mensajes.

Execute. Identifica un programa o un procedimiento que se quiere ejecutar, normalmente un Job se compone de varias tareas (JobSteps) a realizar, cada una de estas tareas debe ir encabezada por esta instrucción EXECUTE.

```
//stepname EXEC PGM=programa
```

PGM identifica el programa (Tarea) a ejecutar.

Data Definition. Identifica cada uno de los conjuntos de datos que intervienen en la ejecución de cada tarea (JobStep), hay que hacer notar que se pueden definir estos conjuntos de datos para cintas, discos e impresoras así como para datos que vengan en la propia corriente de control. El formato general es el indicado a continuación:

```
//ddname DD dsname
```

ddname: es el nombre que se asigna a un dataset.

dsname: es el nombre del dataset definido en la Data Definition.

Ejemplo: //PAYROLL DD DSN=MONTHLY.PAYROLL.DATA

ddname: PAYROLL.

dsname: MONTHLY.PAYROLL.DATA

Formato específico para salidas impresas a impresora o fichero formato impresión.

SYSOUT Data Sets. Es un dataset de salida.

Variantes:

```
//ddname DD SYSOUT=dsn
```

dsn es el nombre del dataset donde queremos colocar los datos.

```
//ddname DD SYSOUT=*
```

El * especifica que deseamos imprimir utilizando la impresora por defecto.

Formato específico para entradas.

SYSIN DataSets. Permiten abastecer a un job con datos de entrada.

```
//SYSIN DD dsname
```

El dsname indica el lugar donde están los datos de entrada.

Variantes:

```
//SYSIN DD DSN=MONTHLY.PAYROLL.DATA
```

Los datos de entrada están en MONTHLY.PAYROLL.DATA

```
//SYSIN DD DATA
```

Indica que los datos vienen a continuación del job.

DataSets para discos y cintas. Su formato es el indicado a continuación:

```
//ddname DD dsname,param1,param2
```

Ejemplo: un dataset se crea en un paso (jobstep, una tarea) dentro de un job, se pasa a otro paso de un job y se borra dentro del mismo job, para ello el dsname debe obligatoriamente comenzar con dos signos, &&, se parte del supuesto ya comentado anteriormente de que un job (trabajo) se compone de varios jobsteps (tareas) y se trata de explicar a continuación cómo es el proceso de los dataset en función de cómo acaba el proceso, normal o anormalmente, y sobre qué se desea hacer con estos ficheros (datasets) en cada paso.

Parámetros de Data Definition.

Veamos una lista de los más importantes:

VOLUME: se utilizan para indicar un disco o una cinta específica donde residen los datos. Los datasets catalogados no requieren este parámetro. Los datasets no catalogados sí requieren este parámetro.

Formato: VOL=SER=aaaaaa (Donde aaaaaa es el número serial de volumen asignado).

DISP – Disposition Parameter: describe el estado del dataset así como el proceso de los datasets tanto en las terminaciones normales como las anormales.

Ejemplos:

DISP=SHR

DISP=(NEW,PASS)

DISP=(OLD,DELETE,KEEP)

El primer parámetro posicional indica:

NEW: el dataset se crea en este jobstep.

OLD: el dataset existe previamente y el jobstep requiere su uso exclusivo.

SHR: el dataset existe previamente y el jobstep NO requiere su uso exclusivo.

El segundo parámetro posicional indica en caso de terminación normal:

DELETE: el dataset se borra en este paso (jobstep).

KEEP: el dataset se conserva.

PASS: el dataset se guarda para un paso posterior y éste decidirá lo que se hace.

CATLG: el dataset se guarda y se crea una entrada en el catálogo.

UNCATLG: se guarda y la entrada en el catálogo se quita.

El tercer parámetro posicional indica en caso de terminación anormal:

DELETE: el dataset se borra en este paso (jobstep).

KEEP: el dataset se conserva.

CATLG: el dataset se guarda y se crea una entrada en el catálogo.

UNCATLG: se guarda y la entrada en el catálogo se quita.

Parámetro UNIT.

Es necesario para especificar la cinta o el disco donde reside el dataset. Los dataset catalogados no requieren este parámetro pero sí los dataset no catalogados así como los dataset nuevos.

Formato: UNIT=aaaaaa donde aaaaaa puede ser alguno de los siguientes valores:

TAPE Cualquier cinta disponible.

TAPEC Cartucho de cinta (Para datasets nuevos o viejos)

TAPECC Cartucho de cinta (Para datasets nuevos)

SYSDA Para discos.

Parámetro SPACE.

Este parámetro se utiliza para asignar espacio a los datasets nuevos.

Format: SPACE=(aaa,(p,s))

Aaa se expresa en CYL (Cilindros), TRK (Tracks), o un bloque de longitud si es pequeño.

P expresa el espacio para asignación primaria que se necesita para el dataset.

S expresa el espacio para asignación secundaria cuando no caben más datos en la primaria.

Parámetro Data Control Block.

Se utiliza este parámetro para definir un nuevo dataset.

Formato: DCB=(LRECL=nnn,BLK.SIZE=nnnnn,RECFM=aaa)

Donde aaa puede tomar alguno de los siguientes valores:

F Longitud fija, desbloqueado.

FB Longitud fija, bloqueado.

V Longitud variable, desbloqueado.

VB Longitud variable, bloqueado.

VBS Longitud variable, bloqueado, extendido (spanned)

U Indefinido.

Donde nnn es la longitud del mayor registro del dataset.

Donde nnnnn es la longitud de cada bloque en el dataset.

COMMENT.

Permite escribir comentarios dentro de un JCL, para ello se escribe después de los dos // un *

Formato: /* texto que se quiere comentar.

DELIMITER.

Permite indicar el final de un JobStep (Tarea).

Formato: /*

NULL.

Es la última instrucción dentro de un Job.

Formato: //

2. LAS TÉCNICAS Y HERRAMIENTAS DE OPERACIÓN AUTOMÁTICA.

Dada la complejidad de las instalaciones actualmente operativas en diversas partes del mundo, se ha desarrollado una serie de herramientas para ayudar a una operación lo más automática posible. Podemos distinguir tres grandes líneas de actuación:

- Automatización del JCL.
- Automatización de los backup's.
- Monitorización y control.

Veamos cada una de estas líneas de actuación:

AUTOMATIZACIÓN DEL JCL.

Entre las herramientas existentes describimos una de las de mayor uso del mercado: JOB/SCAN de la empresa Diversified Software.

Metas con JOB/SCAN.

- Asegurar que su JCL de producción no tenga errores.
- Eliminar chequeos manuales para automatizar la migración de test/producción.
- Estandarizar el formato y el contenido de JCL.
- Automatizar las conversiones SMS.
- Optimizar los recursos utilizados por jobs en batch.
- Corregir errores comunes y violaciones de sus estándares.

La herramienta probada de automatización de JCL.

- Funcionalidad completa y rendimiento superior.
- Escribe sus programas de estándares en lenguajes que conoce y que conforman a SAA, como REXX, COBOL, PL/I o Assembler.
- Instrucciones fáciles de 4GL para efectuar cambios en masa en su JCL, PROC's y librerías de tarjeta de control.
- No hace falta una formación extensiva.

Funciones más importantes de JOB/SCAN.

- JCL de ejecución o PROC's como input.
- Ejecutar con solo un job o con todo su sistema.
- Uso a través de paneles ISPF, macros o ejecución en batch.
- Mensajes de error con texto extendido.
- Ayuda específica relacionada al contexto de cada problema.
- Interfaces con programas de scheduling, control de cambios y gestión de recursos y librerías.
- Compatible con todo el software de los centros de datos.
- Informes sobre el uso del producto.
- Detección automática de la versión de OS con opción de anulación manual.

AUTOMATIZACIÓN DE LOS BACKUP'S.

Vamos a presentar un ejemplo de la vida real, el Centro de Cálculo de la Facultad de Informática pretende dar la posibilidad a los distintos Departamentos del Centro de disponer de un sistema centralizado, cómodo, seguro y fiable, sobre el que poder realizar las copias de seguridad de la información más crítica existente en los servidores de dichos Departamentos.

Todo este sistema se ha montado sobre una plataforma hardware compuesta por un equipo IBM/RS6000 (Raticulin), una torre de discos RAID COMPAREX Storage Subsystem D1000 y un robot de cintas IBM Ultrium Scalable Tape Library 3583-TL.

La torre de discos RAID tiene una capacidad de 84GB. La unidad de cintas dispone de dos bocas y permite cargar hasta 18 cintas, cada una de ellas con una capacidad media aproximada de 100GB.

Con respecto al apartado software, el sistema operativo base es AIX y sobre él se ha instalado la herramienta de IBM Tivoli Storage Manager (TSM). Cada una de las máquinas que quieran utilizar dicho servicio, deberán instalar en las mismas el correspondiente cliente de backup, que suministra el CC.

TIVOLI Storage Manager (TSM) es una herramienta de IBM que permite, entre otras cosas, la gestión, realización y recuperación de copias de seguridad de distintas máquinas de una red, sobre un sistema central o servidor TSM. A dicho servidor TSM se pueden conectar clientes de múltiples plataformas.

Para el funcionamiento de la herramienta es necesaria una parte de servidor y un cliente para cada uno de los componentes a los que se desee conectar. Centrando el tema sobre el componente de Backup/Archive, se pueden encontrar clientes para múltiples entornos, entre otros:

- Microsoft Windows 95, 98 y NT.
- Windows NT DEC Alpha.
- Novell Netware.
- AIX 4.1.
- AIX 4.2, 4.3.
- HP-UX 10.
- HP-UX 11.
- HP Tru64 UNIX.
- HP OpenVMS.
- SCO UNIX 386.
- SUN Solaris 2.6/2.7.

Inicialmente todas las operaciones de backup realizadas por los clientes son volcadas a la unidad de discos; cuando esta unidad supera un determinado umbral de ocupación, la información es enviada al robot de cintas. Esto permite que las operaciones de backup sean totalmente desatendidas. Pero existe una pequeña probabilidad de que las operaciones de recuperación deban ser atendidas desde la sala de operadores.

Dentro del sistema, el administrador tiene la posibilidad de definir diversos parámetros, entre otros:

Número máximo de copias por fichero.

Con este parámetro se indica el número máximo de copias de un fichero que se pueden almacenar. Cada vez que se realiza backup de un cliente, se verifica si ha habido modificaciones en los ficheros, en caso afirmativo se realizará backup del mismo. En el servidor quedarán almacenadas tantas versiones del fichero como se especifique en este parámetro. A estas copias las denominaremos copias extras.

Número de días que se mantienen las copias extras.

Número de días en que se mantiene una sola copia.

Cuando un usuario borra, consciente o inconscientemente, un fichero de su máquina, al realizar el backup y verificar que dicho fichero ya no se encuentra en el cliente, se eliminan del servidor todas las copias extras existentes, conservando exclusivamente la última copia realizada.

Este parámetro especifica el número de días durante los cuales se mantendrá dicho fichero en el sistema para su posible recuperación. Pasado dicho tiempo este fichero será borrado del servidor.

Compresión de la información.

Este parámetro permite que el administrador pueda obligar a los clientes a comprimir la información de forma automática antes de enviarla.

La herramienta TSM permite la realización, tanto de backup (copias de seguridad), como de archive. La finalidad del backup es la de realizar copias de seguridad de la información existente en un equipo, así como la recuperación de ésta en aquellos casos en que pueda ser necesario. El archive se considera como el almacenamiento remoto de una determinada información que no se prevé que vaya a ser necesaria en un período de tiempo, pero que determinados condicionantes obligan a mantenerla.

Se ha diseñado inicialmente este servicio para dar soporte a un máximo de dos o tres máquinas por Departamento, dependiendo siempre del volumen de espacio en disco existente en cada una de estas máquinas, así como del índice medio de modificaciones realizadas diariamente.

Los parámetros indicados anteriormente, junto con las estadísticas diarias que se obtienen del sistema, permitirán dimensionar en cada momento los requisitos hardware necesarios.

Se ponen a disposición de los Departamentos, bajo las especificaciones indicadas por el Centro de Cálculo, los servicios de Backup y de Archive.

Sobre cada uno de los sistemas que quieran conectarse a este servicio, habrá que instalar la parte cliente correspondiente a dicho sistema.

Tras la solicitud por parte de un Departamento, avalada por el Director del mismo, al CC del Servicio Centralizado de Backup, se realizará un estudio de las necesidades del mismo, así como una planificación de la forma en que se va a prestar y recibir dicho servicio.

Dentro de este estudio se especificarán los ficheros o tipos de ficheros de los que se debe realizar backup y de los que no. El espacio máximo de ocupación permitido para cada máquina. En principio, quedan excluidos todos aquellos ficheros pertenecientes al Sistema Operativo de cada máquina, así como el del software correspondiente a aplicaciones y herramientas que puedan ser reinstalables en cualquier momento desde su soporte original. Sí que se contempla la posibilidad de realizar backups de los ficheros que aun siendo del sistema correspondan con información de configuración propia del entorno.

Para cada sistema se asignará un «Cliente» en el servidor de TSM. Cada Departamento deberá nombrar un responsable único del servicio de backup. Este responsable será el conocedor de los usuarios TSM definidos para sus máquinas, así como de las claves de acceso a los mismos.

Desde el CC se definirán las ventanas de tiempo en las que cada máquina podrá comenzar la realización de su backup, por lo cual en los clientes habrá que especificar:

Número de días que se mantienen la copia extra: 15 opción de planificación.

Inicialmente se define una política de backup's, basada en los siguientes parámetros:

- Máximo número de copias por fichero: 2.
- Número de días que se mantiene una sola copia: 30 (caso de borrado).
- Se obliga a todos los sistemas a comprimir la información antes de enviarla.
- Los backups se realizarán en las ventanas de tiempo definidas por el CC.

El Centro de Cálculo se responsabiliza de mantener operativo dicho sistema, exceptuando causas de fuerza mayor y las limitaciones de los medios disponibles para este servicio. Los usuarios del Servicio serán los encargados de verificar que los backups se han realizado correctamente y de forma habitual. Periódicamente desde el CC se emitirán informes de control sobre la utilización del sistema y se intentará dar aviso a los usuarios de los problemas que se hayan detectado.

Tal y como se ha indicado en el apartado anterior las operaciones de backup se realizan de forma desatendida, para ello el robot de cintas contará con las suficientes cintas como para garantizar este servicio. Por otro lado, la recuperación de información de las copias de seguridad puede llegar a ser una operación que necesite de la asistencia de personal. Por ello, antes de iniciar un proceso de recuperación habrá que informar en la Sala de Operadores de este hecho.

MONITORIZACIÓN Y CONTROL.

Soluciones de Gestión y Automatización de Procesos de Negocio. Hoy en día, los rápidos y enormes cambios tecnológicos, así como la globalización, están forzando a las compañías de todos los sectores a rediseñar los mecanismos o formas en la que conducen su negocio, evolucionando de operaciones independientes y cerradas, a un marco electrónico, dinámico e integrado que garantice la colaboración con clientes, proveedores y colaboradores.

Con el objetivo de competir eficazmente, las organizaciones están esforzándose en mejorar su eficiencia, incrementar la colaboración, mejorar la atención a sus clientes, reducir el tiempo de desarrollo de nuevos productos/servicios y su salida al mercado, así como ser capaces de adaptarse rápida y flexiblemente a los cambios impuestos por el mercado.

Con la utilización de soluciones de gestión y automatización de procesos de negocio, las organizaciones obtienen mayor flexibilidad y capacidad para diseñar, desarrollar e implantar procesos críticos del negocio a través de redes, sistemas y plataformas heterogéneas, permitiendo:

- Crear una visión de procesos extremo a extremo dentro de la corporación.
- Integrar los procesos de negocio y las aplicaciones a través de múltiples empresas.
- Monitorizar los procesos del negocio para analizar su eficiencia.

- Responder a los cambios del negocio mediante modificaciones rápidas y sencillas sobre los procesos.
- Automatizar la totalidad de los procesos de la empresa, desde los críticos hasta los administrativos.
- Aprovechar las oportunidades de comercio electrónico.

Las organizaciones que han comprendido los beneficios del negocio electrónico (eBusiness) necesitan un nivel más profundo de integración que se extienda más allá de sus aplicaciones e infraestructuras de datos y que permita que las personas y procesos de negocio interactúen con su infraestructura.

En esta situación o demanda de mercado aparecen términos que comúnmente han sido asociados a soluciones Workflow, tales como procesos de negocio, personas e integración, pero sin embargo surgen nuevas necesidades que se sitúan en otro nivel y que en la mayoría de los casos van asociadas con la capacidad de creación de una infraestructura de integración robusta y extensible. Estas necesidades están siendo cubiertas a través de soluciones denominadas EAI (Enterprise Application Integration), cuyo objetivo es el de integrar múltiples aplicaciones heterogéneas, que utilizan tecnología incompatible y que necesitan permanecer gestionadas de forma independiente. Estas aplicaciones pueden incluir bases de datos, aplicativos Host, ERPs, CRMs, aplicaciones específicas, etc.

Soluciones BMP.

Las soluciones EAI están enfocadas principalmente a la integración sistema a sistema y al intercambio de datos, pero no provee de las capacidades de gestión de procesos extremo a extremo y de interacción humana que la mayoría de los procesos de negocio requieren.

De igual modo, las soluciones Workflow están enfocadas a encaminar formularios y ficheros para aprobación, pero no proveen las capacidades necesarias de integración y/o gestión de transacciones entre diferentes aplicaciones clientes y sistemas back-end.

Una solución BPM es aquella que engloba lo mejor de Workflow y EAI, es decir, un entorno de integración para entornos heterogéneos, con un motor de procesos robusto, escalable y con gran capacidad de procesamiento automático, que permita definir, gestionar y medir los procesos críticos de la empresa.

Hoy en día, las soluciones BPM permiten diseñar, desarrollar e implantar procesos críticos de negocio (misión crítica) a través de redes heterogéneas, sistemas operativos y plataformas cliente. Proveen el nivel de inteligencia que incorporan los sistemas EAI (procesamiento automático), así como sofisticada funcionalidad Workflow (procesamiento manual).

Las soluciones BPM monitorizan y gestionan los procesos con el objetivo de que sean altamente automatizables, altamente disponibles, altamente escalables y altamente eficientes, haciendo uso para ello de una combinación de balanceo de carga, tecnología de brokers, monitorización de transacciones, gestores de recursos y procesamiento distribuido.

Soluciones Workflow.

Workflow es una solución tecnológica que ayuda a las organizaciones a definir cómo automatizar y gestionar una actividad particular o un proceso. Considera criterios cómo las reglas deben ser segui-

das, los roles individuales que deberían participar en el proceso y cómo los datos deberían ser recogidos y encaminados. Workflow tiende a ser muy interactivo, demandando intervención manual para iniciar, intervenir y aprobar pasos en los procesos.

En la actualidad existe un gran abanico de productos bajo el paraguas de Workflow, pero con una gran diferenciación en cuanto a funcionalidad y plataforma tecnológica de trabajo:

Workflow de Producción: herramientas de Workflow robustas, multiplataforma, escalables e independientes de otras soluciones o sistemas, con capacidad de automatizar procesos críticos de la empresa, donde primen la intervención manual sobre la automática.

Workflow Específico de Aplicación: herramientas de Workflow robustas, multiplataforma y escalables pero asociadas a otras soluciones o sistemas, tales como SGBDR (Sistemas de Gestión de Bases de Datos Relacionales), ERP, CRM, etc.

Workflow Administrativo, Adhoc y asociados a Gestores Documentales: herramientas de Workflow de todo tipo.

Workflow asociados a Mensajería y Trabajo en Grupo: aplicaciones que dotan a una herramienta de mensajería o trabajo en grupo, de funcionalidad para crear pequeños procesos o ciclos de aprobación dentro de su propia infraestructura, es decir, haciendo uso de la bandeja de correo como lista de actividades.

En la mayoría de los casos no dispone de un motor de procesos claramente definido, sino que se trata de un pequeño conjunto de funciones programáticas que permiten orientar el «recorrido» de formularios o actividades.



