TEMA 2

SOPORTE LÓGICO

- Programas e instrucciones.
- Introducción.

Una instrucción es un conjunto de símbolos(que usualmente son caracteres) que representan una orden de operación o tratamiento de información para la computadora. Las instrucciones suelen realizarse con datos o actuar sobre estos. Un programa es un conjunto ordenado e instrucciones que se suministran al ordenador y le indican la tarea a realizar.

Las instrucciones se forman con elementos o símbolos de un repertorio determinado y se construyen siguiendo unas reglas precisas. Al conjunto de símbolos y reglas sintácticas con las que se redactan los programas, se le denomina lenguaje de programación. Los circuitos de la computadora solo pueden leer instrucciones formadas por bits 0 y 1, que conforman un conocido lenguaje llamado maquina. Estos bits están agrupados en bloques o campos. En todas las instrucciones maquina existe un bloque que contiene el código de operación(un conjunto de bits que identifican la operación a realizar), y en la mayoría de ellas existe un bloque de dirección que contiene información para acceder al dato sobre el que actúa el bloque de operación.

Las instrucciones se pueden clasificar en los siguientes grupos:

- De transferencia de datos de E/S.
- De calculo o tratamiento(aritmético−lógicas).
- De bifurcación o ruptura de frecuencia, que permiten alterar el orden de ejecución de las sentencias.
- De control.

Atendiendo a la estructura podemos clasificar las instrucciones maquinas en:

- De tres operandos. Al código de operación la siguen tres operándos, los dos primeros son las direcciones de los datos con los que se va a operar, y el tercero corresponde a la dirección de memoria donde se va a guardar el resultado.

direcciones

- Con dos operandos. Al código de operación le siguen dos operaciones de memoria que apuntan a las posiciones que contienen los datos que van a intervenir en la operación actuando una de ellas como receptora del resultado de la operación.

- Con un operando. Se utiliza generalmente en computadoras cuya arquitectura funciona con filosofía de acumulador (Ej: ODE), al código de operación le sigue la dirección de uno de los operandos, en el acumulador se deposita el resultado de la operación.
- Sin operandos. Se utilizan con arquitecturas con filosofía de pila, el sistema mantiene una serie de punteros para la gestión de la pila y tanto un operando como el otro se extraen de la pila y el resultado se vuelve a guardar en la pila.

Normalmente los operandos que van a intervenir en la operación están correlativamente en la pila.

- Métodos de direccionamiento.

Es el modo que utiliza una instrucción para indicar la posición de memoria del dato o los datos que van a intervenir en la misma. Normalmente en una instrucción maquina se suelen utilizar algunos de los siguientes métodos de direccionamiento:

- Direccionamiento inmediato. En este método el dato que interviene en la instrucción forma parte de la propia instrucción no necesita por tanto ningún acceso a memoria para acceder al dato.
- Direccionamiento directo. Con el código de operación se especifica la dirección de la posición de memoria que va a intervenir en dicha operación, se necesita un acceso a memoria para trasladar el dato desde la memoria interna hasta el bloque de operaciones(Ej: ODE).
- Direccionamiento indirecto. Con el código de operación se especifica la dirección de la posición de memoria que contiene la dirección del dato, a esta dirección se le denomina dirección intermedia, para acceder al dato se necesita al menos 2 accesos a la memoria interna, pero este modo permite direccionar un mayor número de posiciones de memoria, porque utiliza todos los bits de la palabra para codificar la dirección.
- Direccionamiento relativo. Con el código de operación se especifica la posición en la memoria a partir de una dirección base que se haya almacenada en un registro especial. De esta manera se posibilita el acceso a un conjunto de posiciones consecutivas a partir de esta dirección base.

instrucción

Dir (n + k)

+

k= Dirección base o de referencia

1.3 Ciclo de una instrucción.

Como sabemos un programa para ser ejecutado debe estar en la memoria interna, de la cual la CPU va extrayendo una a una las instrucciones que marcan la secuencia del programa y que le indica la tarea que va a realizar.

Se denomina ciclo de instrucción al conjunto de acciones que se llevan a cabo para ejecutar una instrucción. En el ciclo podemos distinguir dos fases:

- De búsqueda.

En ella se transfiere la instrucción que corresponde ejecutar desde la M.I. hacia la unidad de control. Para ello se realizaran los siguientes pasos:

- La unidad de control envía una micro−orden para que el registro contador del programa que contiene la dirección de la siguiente instrucción ha realizar sea transferido al registro de la dirección de memoria(RDM).
- La posición de memoria que figura en el RDM es utilizado por el selector para transferir su contenido(dato) al registro de intercambio de memoria(RIM).
- El contenido del RIM(la instrucción) se transfiere al registro de instrucción de la unidad de control.
- El decodificador de la unidad de control interpreta la instrucción que acaba de llegar al RIM y el secuenciador genera las micro−ordenes pertinentes.
- El registro contador de programas(CP) se auto−incrementa en 1 utilizando para ello los circuitos de la UAL, apuntando de esta manera a la siguiente instrucción. Si la instrucción fue de ruptura de secuencia el RCP se carga con la dirección de la instrucción a la que debe saltar.

- De ejecución.

Consiste en llevar a cabo todas las acciones que lleva la instrucción, se realiza siguiendo los siguientes pasos para dos operandos, sino es de dos operandos algunos pasos no se hacen.

- Se transfiere la dirección del primer operando desde el RI hasta el RDM.
- Se activa el selector, y extrae el contenido de la posición de memoria cuya dirección se haya en el RDM y la transfiere al RIM.
- Se transfiere el contenido del RIM hacia el primer registro de entrada de la UAL.
- Se transfiere el contenido del segundo operando del RI al RDM.
- El selector lo extrae y lo transfiere al RIM.
- Se transfiere el contenido del RIM hacia el segundo registro de la UAL.
- El secuenciador emite una micro−orden a la UAL para que realice la operación correspondiente con los datos de los registros de entrada y el resultado se mete en el acumulador.
- Se transfiere desde el RI al RDM la dirección de memoria donde ha de depositarse el resultado.
- El contenido del acumulador pasa al RIM, se activa el selector que deposita su contenido en la dirección a la que apunta el RDM.

2. Lenguajes de programación.

Para que una computadora funcione es necesario que un programa le indique que es lo que tiene que hacer por medio de instrucciones que forman una secuencia ordenada. Estas acciones se pueden clasificar en sentencias imperativas (que indican explícitamente una acción a realizar) y aclarativas (informan sobre una circunstancia del programa, el tipo de variable).

Un programa y sus sentencias se redactan con unos símbolos determinados(el alfabeto y los signos de puntuación) y de acuerdo con unas reglas que determinan la gramática del lenguaje.

Tradicionalmente se ha clasificado el lenguaje de programación en tres niveles:

- Maquina. Es el que entiende la computadora.
- De bajo nivel o ensambladores.
- De alto nivel.

2.1. El lenguaje Maquina.

La característica principal de este lenguaje es que resulta muy engorrosa su utilización y que obliga al programador a conocer perfectamente la estructura física de la computadora, ya que diferentes plataformas tendrán diferentes lenguajes maquina. No obstante presenta la ventaja de ser ejecutable por la computadora no necesitando ningún lenguaje previo para entenderlo.

En un lenguaje maquina:

- Las instrucciones están formadas por cadenas de 0 y 1 pudiéndose dar a la computadora un código intermedio(octal o hexadecimal).
- Los datos se le dan a la computadora mediante sus direcciones de memoria, el programador tiene que saber donde están los datos y las instrucciones para que no se solapen entre si.
- El repertorio de instrucciones suele ser muy reducido.
- Los programas resultan poco legibles y poco elásticos ya que el formato de las instrucciones es rígido.
- Los programas son poco transferibles ya que las instrucciones están íntimamente ligadas a la arquitectura de la computadora.

2.2. Lenguajes ensambladores.

Suponen un paso intermedio entre el lenguaje máquina directamente interpretable por la computadora y los lenguajes de alto nivel mucho más próximos al lenguaje natural. Las instrucciones en ensamblador están formadas por códigos nemotécnicos, es decir por un conjunto de caracteres (normalmente 2 ó 3) que identifican la acción que realiza una instrucción. Porque para sumar de una ADD normalmente una instrucción en ensamblador corresponde a una o varias instrucciones máquina. Este lenguaje permite hacer referencia a los datos por medio de direcciones de memoria simbólicas en vez de direcciones absolutas que se usan en lenguaje máquina.

También permite la inserción de comentarios en el código del programa para facilitar la legibilidad del mismo. Para que una computadora pueda ejecutar los programas primero hay que traducirlos por un programa traductor también llamado ensamblador.

Una versión más evolucionada de los ensambladores son los macroensambladores. En los cuales la mayoría de las instrucciones de su repertorio correspondiente a varias instrucciones máquina, llamadas también macroensambladores.

2.3. Lenguajes de alto nivel.

No obligan al programador a conocer la estructura interna de la computadora. Usan sentencias con una semántica parecida al lenguaje natural(normalmente Ingles).

Las principales características son:

- Para construir las instrucciones se usan caracteres alfabéticos, numéricos y especiales. El programador puede referirse a un dato mediante un nombre arbitrario normalmente significativo(Ej: int sueldo), (son las variables, constantes, campo de registros, etc..).
- El repertorio de instrucciones es muy amplio, siendo las reglas gramaticales para su construcción muy flexibles.
- Apenas dependen de la máquina, no obstante existen lenguajes más trasladables que otros. De tal efecto se han creado organismos encargados de crear versiones estándar de los mismos lenguajes(como el ANSI).
- Los lenguajes de alto nivel permiten la inclusión de líneas de comentario que facilitan la legibilidad del código.
- Un programa en lenguaje de alto nivel puede ser traducido por un programa traductor, compilador o interprete para poder ser ejecutado.

Lenguaje de alto nivel Ensambladores Lenguaje maquina(octal)

A=B+C LDA 0,4,3 021404

LDA 2,3,3 021403

ADD 2,0 143000

STA 0,5,3 041405

2.4. Programas traductores.

Los traductores son programas que traducen o trasladan un programa escrito en ensambladores o lenguaje de alto nivel(en definitiva en un lenguaje simbólico) a un lenguaje maquina directamente interpretable por una

computadora(lenguaje maquina). Al programa inicial sobre el que actúa el traductor se le denomina programa objeto.

Básicamente existen dos tipos de traductores; los interpretes y los compiladores.

2.4.1 Compiladores.

Un compilador traduce un programa fuente escrito en lenguaje simbólico(fichero fuente) a un lenguaje intermedio(posiblemente ensambladores) o lenguaje maquina que pasa a formar un fichero objeto.

La compilación consta de dos etapas principales que a veces no están claramente diferenciados, son:

Análisis del programa fuente.

Síntesis del programa objeto.

En estas etapas podemos distinguir las siguientes fases:

- Análisis léxico−gráfico.

Consiste en descompensar el programa fuente en sus elementos constituyentes: palabras reservadas, símbolos de operaciones, identificadores, comentarios, etc...

El analizador léxico−gráfico o escáner debe identificar el tipo de cada elemento ídem generando tablas de símbolos a demás de convertir números y constantes del código de E/S a su representación interna.

- Análisis sintáctico.

Identifica las estructuras(expresiones, sentencias aclarativas, etc...) que componen el programa. La sintaxis de un lenguaje especifica como han de escribirse las sentencias. El analizador sintáctico descompone estas sentencias en árboles que identifican las palabras reservadas que identifican los símbolos, las estructuras, las palabras reservadas, etc...

- Análisis semántico.

La semántica de un lenguaje de programación se refiere al significado que se da a las distintas construcciones sintácticas, en esta fase empieza a generarse el código objeto completándose las tablas generadas en fases anteriores, también se insertan las instrucciones correspondientes a las macros y se ejecutan aquellas sentencias que se han de llevar a cabo en tiempo de compilación.

- Optimizaciones.

El analizador semántico generalmente crea un programa en código intermedio utilizando por ejemplo la notación polaca que es una forma de evaluar las expresiones. En esta fase de optimización se mejora el código intermedio analizándose el programa en su conjunto.

For x=1 to 1000 a=28

A=28 for x=1 to 1000

?x ?x

next next

- Generación del código objeto.

En esta fase se genera el fichero objeto que puede ser directamente ejecutable por el ordenador(extensión .exe) hay otros ficheros como .com o necesitas una etapa previa llamada linkeditacion, en este caso el fichero generado es . obj.

2.4.2 Interpretes.

Un interprete es un programa que traduce un programa escrito a alto nivel a código interpretable por la computadora, sentencia a sentencia generando varias instrucciones maquina por cada instrucción en alto nivel, no se crea ningún fichero objeto y la ejecución se realiza sentencia a sentencia inmediatamente después de ser producida, las fases de análisis y las de optimización se realizan en el contexto de la sentencia y no del programa entero, por tanto se ejecuta más rápido, pero cada vez que hay que ejecutarlo hay que volver a interpretarlo. Los mensajes de error son suministrados al programador en el mismo instante al que se produce, pudiendo abortar el proceso, esto permite la corrección automática del error por eso se dice que desde el punto de vista pedagógico los lenguajes interpretados son más importantes que los compiladores para aprender a programar.

3. Organización de los datos.

Entendemos por dato cualquier valor manipulado por la computadora, puede ser un carácter lógico del teclado, una información almacenada en un disco, un valor en memoria interna, etc...

Son datos tanto las constantes(no cambian de valor) como las variables(identificadores que se pueden variar su valor). Así mismo son datos las información externa al programa a la que se puede acceder por algún procedimiento, ya sea información grabada en algún soporte o procedente de algún dispositivo.

3.1 Tipos de datos.

Un tipo de dato es el conjunto de la transformación magnitud−dato y de las operaciones internas para ese conjunto.

3.1.1 Datos elementales.

- Datos de tipo entero. Es una representación del conjunto de los números

normalmente englobados bajo el denominador integer(int). La transformación consiste en representar el número en binario y almacenarlo en un número fijo de bits. El número máximo para representar es el 2n, normalmente los lenguajes de programación restringen el administrador integer a aquellas magnitudes comprendida entre los valores

[−32768,32767], ya que se utilizan 16 bits para su codificación. Cualquier entero que supere ese rango será englobado en el tipo longint, si el lenguaje de programación lo soporta, y sino será tratado como real. Nota: existen lenguajes de programación que admiten restricciones de los enteros como el tipo de dato byte, se reserva un solo byte para la codificación. En algunos existe el word.

- Datos de tipo real. Es una representación del conjunto de los números reales, esencialmente la transformación realizada consiste en expresar el número de la forma siguiente N =m * be, donde N es el número a representar, b es la base utilizada en la representación del exponente, e es el exponente y m es la mantisa. Este tipo de representación denominada como flotante permite la representación de

un número muy grande o muy pequeño.

Cada dato de tipo real representa un conjunto infinito de números reales.

- Datos de tipo lógico o booleano. Pueden representar magnitudes true o false (verdadero o falso). Sobre este tipo pueden actuar los operadores lógicos and, or, xor, etc...

Se denomina expresión o relación lógica a toda aquella expresión formada por 2 operandos pertenecientes a un mismo tipo de ordenador y un operador de relación(>,<,<=,>=,=.<>), el resultado de evaluar la relación será un dato de tipo booleano.

Ej: a=5 b=10

a>b falso

a<b verdadero

- Datos de tipo carácter. Representan conjuntos finitos y ordenados de caracteres, para un sistema dado normalmente el conjunto de los tipos char estarán formado por aquellos caracteres pertenecientes al alfabeto utilizado en el código de E/S(normalmente ASCII). La mayoría de los lenguajes poseen funciones de conversión de caracteres al código que lo representa y viceversa.

chr (65) A

asc (A) 65

- Datos de tipo enumerado. Se definen dando un conjunto de valores no se trata de un tipo normalizado y en cierto modo podemos considerarlo como una clase de tipo de datos a las que pertenecerán por enumeración de sus elementos.

Ej: type dias.laborables=(lunes, martes, miércoles, jueves, viernes, sábado, domingo)

En la mayoría de los lenguajes que soportan este tipo las definidas no se pueden utilizar para capturar datos de Entrada y verdaderamente tampoco de salida.

Los datos de tipo numerado se crean mediante software. Internamente los datos de tipo numerado están representados por valores enteros.

- Datos de tipo subrango. Son subconjuntos definidos a partir de tipos de datos elementales ordinales(tipo carácter, enumerado, entero, entero largo, etc...). un tipo de dato subrango puede tomar cualquier valor comprendido entre el valor mínimo y el máximo que define su subrango.

Ej: var x:dias_laborables;

Y:1.....5;

3.1.2 Estructuras de datos.

Los tipos de datos elementales se pueden utilizar para construir tipos más elaborados o tipos estructurados. Una estructura de datos esta compuesta por una serie de datos elementales y alguna relación existente entre ellos. Una estructura puede ser homogénea si todos los datos elementales con los que se forman son del mismo tipo(cadena) o heterogénea si al menos uno de ellos es distinto(record). Una estructura se dice que es

dinámica si la memoria que necesita se asigna de forma dinámica, al contrario se trata de estructura estática, que es aquella a la que se asigna memoria cuando se traduce su definición y ya no se puede modificar esta sección hasta que termine la ejecución del programa.

- Formaciones, arrays o vectores:

La formación o arrays es la estructura de datos más usual con permiso del tipo cadena y esta soportado por la mayoría de los lenguajes de programación, se trata de una estructura formada por un conjunto de datos del mismo tipo, cada uno de los cuales esta asociado a un índice que especifica la posición relativa del elemento respecto a la estructura a la que pertenece, los índices sin siempre del tipo subrango.

Generalmente al número de índices del arrays se define como dimensión de un arrays, esta viene dada por el número de índices, y el número del arrays viene dado por la multiplicación de los valores máximos de los índices.

5 elementos 10 elementos 20 elementos 1dimensión 2 dimensiones 3 dimensiones

Ej: Pascal Visual Base

var x:array [1....5] of integer; din x(5) as integer

y:array [15, 14] of real din y(5,2) as single

Cada elemento dentro de una formación viene determinado por el nombre de la información y la posición que ocupa a cada una de las dimensiones de la misma.

- Cadena de caracteres.

Una cadena de caracteres o string es una cadena. Una cadena puede ser cualquier formación ordenada a partir de caracteres. Normalmente los lenguajes de programación suministran funciones de manejo de cadenas que nos permiten la:

Caracterización(normalmente con el generador aritmético + o &).

Extracción(left, rigth, mid).

Comparación(normalmente con el operador =).

Localización de sus cadenas en una cadena(instrucciones en Basic y posteriormente en Pascal).

- Registros.

Un tipo de datos registro es el que está formado por yuxtaposición de elementos que contienen información relativa a un mismo ente. Los tipos de datos de estos elementos no tienen porque ser del mismo tipo, a cada uno de los elementos que conforman el registro se le denomina campo, normalmente se utiliza el tipo registro para implementar el tipo fichero o del tipo file.

- Listas dinámicas.

Esta formada por un número variable de elementos de un mismo tipo ordenado según una secuencia lineal, cada elemento excepto el primero tiene un predecesor en la lista y todos salvo el ultimo tiene un sucesor en la lista, la lista dinámica es no direccionable(para su manejo se utilizan punteros y a demás se le asigna memoria

de forma distinta.

Single (4 bytes)

Double (8 bytes)

Las operaciones esenciales que suelen utilizar una lista dinámica son inserción, eliminación, localización de un elemento, si la lista esta ordenada. Los algoritmos de inserción, eliminación y localización son más complejos.

Para almacenar una lista dinámica en memoria se utilizan punteros. Un puntero es un dato que almacena una dirección de memoria para la información y para el puntero que va a señalar al siguiente elemento de la lista.

Un caso particular de lista es aquel en el que se insertan o eliminan elementos solo en un extremo de la misma. Se denominan lista o lifo, o pila lifo a aquella que se maneja bajo la premisa de que el ultimo elemento en entrar en la lista es el primero en ser eliminado(last in first out). Se denomina lista fifo o cola en la que el primer elemento que entra es el primero que sale.

- Árboles.

Un árbol es una estructura de datos formada por elementos de varios tipos denominados nodos relacionados de tal modo que pueden descomponerse en un nodo llamado raíz del cual penden un conjunto finito de objetos de tipo árbol denominados sub−árboles. Se denomina hijo a cada uno de los nodos que dependen de uno dado denominado padre, se denomina grado de un nodo al número de sub−árboles que sustenta un nodo. Se denomina orden de un árbol al mayor de los grados de sus nodos. Un árbol es una estructura dinámica de datos que se implementa en memoria interna utilizando punteros, si el orden del árbol es 2 es binario.

Raíz

Rama

Nodo padre

- Ficheros.

Un fichero(file) es un conjunto de información del mismo tipo tratada como una unidad de almacenamiento y organizada de forma estructurada para la búsqueda de un dato individual. Un fichero está compuesto de registros homogéneos que contienen información de un mismo tema. La vida de todo fichero comienza normalmente con una sentencia open y termina con una sentencia close en una sesión de trabajo. Las operaciones básicas que se pueden utilizar con el fichero son inserción de un nuevo registro(alta), eliminación de un registro determinado(baja), acceso a un dato determinado(consulta), edición de un registro determinado(modificación).

El sistema es el encargado de transportar cada vez que se accede al dispositivo que se ha almacenado una cantidad fija de información que depende de las características físicas del dispositivo. Esta información se denomina bloque o registro físico en contraposición al registro lógico que es cada uno de los entes que componen el fichero. En un bloque puede haber varios registros lógicos o puede darse el caso de que un registro lógico ocupe varios físicos. Un factor importante en el diseño de un fichero es la longitud del bloque o factor de blocaje. Que se define como el número de registros del fichero que entran en un bloque, cuanto mayor sea este menor número de accesos se realizaran al dispositivo, el sistema operativo también es el encargado de transformar las direcciones lógicas de los registros (normalmente la posición relativa que ocupa el registro dentro del fichero) en direcciones físicas que son aquellas que a través de las cuales se halla el

registro del dispositivo. El fichero es una estructura de datos externa al sistema que se halla almacenada en un dispositivo de memoria auxiliar. El programa se comunica con el fichero a través de una porción de memoria interna que actúa como buffer del fichero. El tipo de dispositivo condiciona los tipos de organización de los registros del fichero. Se denomina organización de un fichero al método que se utiliza para implementar los registros de acuerdo a unas normas y a unas posibilidades de acceso a los mismos.

Los principales tipos de organizaciones son:

- Secuencial. Los registros aparecen físicos uno detrás del otro y en orden. El acceso a cada uno de los registros obliga a acceder a los que le preceden, no se admiten inserciones entre dos registros existentes, como consecuencia las nuevas inserciones se colocan al final del fichero.
- Organización relativa. Permite el acceso aleatorio a los registros indicando la posición relativa que ocupa el registro dentro del fichero. Por tanto es una organización que se implementa en dispositivos de acceso aleatorio.

Esta posición relativa se conoce como clave relativa o relativa key. La transformación de las posiciones relativas en direcciones físicas del dispositivo la realiza el sistema operativo.

- Organización directa. Consiste en utilizar un algoritmo de conversión que transforma un campo clave de registro(Ej: código socio) en una dirección física en el dispositivo, el acceso es directo y se implementa en dispositivos de acceso aleatorio. No obstante lo que se suele hacer es un algoritmo de conversión de un campo de registro a una operación relativa dejando al sistema operativo que realice la transformación de esta posición relativa a la dirección física por tanto en la memoria de los lenguajes de programación la organización relativa subyace en la directa.

Ej:

122−451

Algoritmo Posición relativa Sistema operativo Dirección física

- Organización indexada. Utiliza un índice de ubicación de los registros que normalmente relaciona un campo del mismo(campo clave) con una dirección física, este índice puede estar implementado dentro del propio fichero de datos o en un fichero a parte (fichero índice) normalmente los lenguajes de programación permiten la existencia de más de un código clave, pero todos obligan tener al menos una clave univoca para cada registro(sin duplicados). Las claves secundarias si pueden tener duplicados. El acceso a los registros es aleatorio aunque no tiene porque ser aleatorio a la búsqueda de índice(normalmente se aplica un algoritmo basado en la búsqueda dicotómica).

4. Introducción a los sistemas operativos. (Muy importante)

4.1. Funciones y objetivos de los sistemas operativos.

U sistema operativo es un programa o conjunto de ellos que controlan la ejecución de los programas de aplicación de los usuarios y que actúa como interfase entre estos y el hardware de la computadora bajo las premisas de:

- Eficiencia. Permite que los recursos de un sistema informático sean aprovechados de forma eficiente, aumentando el rendimiento global de la computadora, se puede decir que el sistema operativo es un programa de control que se encarga de asignar y administrar los recursos de Hw(tiempo de CPU, espacio en memoria, espacio en disco, etc...).
- Comodidad. Una de las misiones de los sistemas operativos es facilitar el uso de la computadora de la

forma más cómoda posible, en esta encontramos todos aquellos sistemas operativos basados en ventanas desde el punto de vista de la interfase. Respecto al funcionamiento interno el sistema operativo por ejemplo se encarga de liberar al usuario de la obligación de conocer la estructura interna de la computadora y así en una operación de E/S son los programas de control de gestionar el dialogo entre la CPU y los dispositivos devolviendo las características a unos y a otros.

- Capacidad de evolución. Un sistema operativo debe construirse de forma que permita el desarrollo efectivo, la verificación y la introducción de nuevas funciones en el sistema sin interferir en los servicios que brinda, en base a los nuevos tipos de Hw y las actualizaciones que proporcione la empresa desarrolladora.

4.2. El sistema operativo como interfaz usuario−máquina.

Si tuviéramos que programar nuestras aplicaciones en código maquina controlando cada función que realiza el Hw la programación se convertiría engorrosa y compleja, para que esto no ocurra existen programas del sistema operativo que se encargan del control de E/S, de la gestión de archivos, de la ayuda de creación de otros programas de aplicación, etc...

Pirámide por capas de los elementos de un sistema informático atendiendo a su relación.

El sistema operativo oculta al programador los detalles de Hw y le proporciona una interfaz cómoda para utilizar el ordenador, por tanto actúa como mediador facilitando al programador y a los programas el acceso y el uso de los recursos que ofrece el sistema, de forma resumida un sistema operativo ofrece servicios en las siguientes áreas.

- Creación de programas. Editores y depuradores ayudan en su tarea al programador, normalmente estos servicios son proporcionados por programas de utilidad que no pertenecen realmente al sistema operativo, pero son accesibles a través de él.
- Ejecución del programa. Las instrucciones y los datos se han de instalar en memoria interna para ser procesados y los dispositivos de E/S han de ser inicializados, de esta tarea se encargan los diferentes módulos del sistema operativo.
- Acceso a los dispositivos de E/S. Cada dispositivo de E/S requiere un conjunto especifico de instrucciones de instrucciones para su funcionamiento. El sistema operativo tiene en cuenta estos detalles de modo que el programa debe pensar solo en términos de escrituras y lecturas simples.
- Acceso a archivos de datos. Es el sistema operativo el que se encarga de blindar los mecanismos de acceso y control de los archivos haciendo transparente al usuario la naturaleza del dispositivo, el método de almacenamiento, y la codificación de los datos.
- Acceso al sistema. El sistema operativo gestiona al ordenador como un todo y blinda protección de los recursos específicos, a los datos, etc... ante usuarios no autorizados, e incluso puede resolver conflictos en la propiedad de estos recursos.
- Detección y respuesta a errores. Durante el funcionamiento de un sistema se pueden producir errores externos o internos en el Hw(errores de memoria, etc...) y de Sw(desbordamiento de una tarea, acceso a direcciones de memoria prohibida, etc...), en cada caso el sistema operativo debe dar una respuesta que elimine la condición de errores con el menor impacto posible, esta respuesta puede ir en la línea de terminar con la causa que lo produjo, por ejemplo finalizando el programa en cuestión, reiniciar la operación que produjo el error o simplemente informar del error producido todo ello intentando que se produzca el menor impacto posible.
- Contabilidad. Todo sistema operativo debe llevar registros estáticos de la utilización de los diferentes recursos, así como de sus rendimientos, esta información es de suma utilidad para el desarrollo de nuevas versiones que aumentan el rendimiento global del sistema.

4.3. El sistema operativo como administrador de recursos.

Una definición típica de computadora alude a un conjunto de recursos utilizados para el traslado, mantenimiento, procesamiento y almacenamiento de datos. El encargado del control de estos recursos es el sistema operativo pero se trata de un controlador algo peculiar, ya que normalmente se piensa en un mecanismo de control como algo exterior a lo que se pretende controlar y sin embargo el sistema operativo:

- Es un conjunto de programas que necesitan ejecutarse de la misma forma que los programas de aplicación, es decir necesita estar en memoria interna y que la CPU lo atienda.
- En ocasiones el sistema operativo pierde el control de la computadora y depende de la CPU para recuperarlo.

El sistema operativo es un programa ,más que de instrucciones al procesador, la diferencia radica en estas instrucciones que es la de dirigir al micro en el uso de los recursos y controlar la ejecución de las aplicaciones, para que esto se pueda realizar el micro debe poder interrumpir la ejecución del sistema operativo para pasar a ejecutar instrucciones de los programas de aplicación, pero ello obliga a que parte del sistema operativo siempre aparezca en memoria interna(es el núcleo o kernel) e incluye las funciones vitales del sistema operativo y algunas de las más utilizadas. El resto de la memoria esta ocupada por las aplicaciones del usuario y los datos. La asignación de este recurso de las memorias la realiza el modulo gestor de la memoria en conjunción con el Hw de gestión de memoria del micro, a demás el sistema operativo decide cuando se pueden utilizar un dispositivo de E/S y controla el acceso a los archivos, también hay que en cuenta que el micro es un recurso más del sistema y es el sistema operativo el que se encarga de asignar los tiempos de CPU según los algoritmos de multiprogramación que utiliza.

4.4. Componentes de un sistema operativo.

Los sistemas operativos se organizan en capas entorno a un núcleo principal, cada una tiene una determinado función siendo el kernel la capa con mayor prioridad esto es así porque las operaciones básicas del núcleo son aquellas que tienen que ver directamente c los componentes físicos del ordenador, el resto de las capas se disponen sobre este núcleo acercándose cada vez más al usuario.

Esencialmente un sistema operativo se puede dividir en los siguientes niveles:

- Núcleo o kernel. Es la primera capa y la de mayor prioridad y se encarga de funciones básicas del sistema tales como la gestión de la memoria o el control de las operaciones de E/S, un concepto que esta muy de moda es el de micro−núcleo o micro−kernel que se refiere a cada una de las partes en las que se divide un núcleo diseñado en forma modular para hacerlo más flexible y fácil de ampliar.
- Nivel ejecutivo. Se encarga del manejo de sistemas de archivos y la gestión de los procesos.

− Nivel supervisor. Interpreta los mandatos que se le dan al sistema operativo.

- Nivel usuario. Controla o suministra las herramientas de desarrollo(interpretes, compiladores,etc...) y administra los procesos que permiten la comunicación con los usuarios(en Linux el interprete de comandos).

Esta división no es la misma en todos los sistemas por ejemplo en Windows NT los niveles son:

- Capa de abstracción de Hw(HAL). Esta capa establece una correspondencia entre ordenes y respuestas genéricas de Hw y aquellas que son propias de plataformas especificas (Intel X−86, Alfa de digital, etc...). la HAL hace que el bus del sistema de cada maquina, el controlador de DMA, el controlador de interrupciones, el reloj del sistema y el modulo de memoria parezcan los mismos para el núcleo en cualquier plataforma.
- Núcleo. Consta de los componentes más usados y fundamentales del sistema operativo, por ejemplo administra la planificación y el cambio del contexto, la gestión de las interrupciones y las

excepciones, y la sincronización de multiprocesadores.
- Subsistemas. Incluye módulos con funciones especificas que se basan en los recursos proporcionados por el núcleo.
- Servicios del sistema. Proporciona una interfaz al Sw para el modo usuario.

En el sistema operativo Unís el Hw básico está rodeado por el Sw del sistema operativo. Normalmente se denomina a todo el sistema operativo núcleo, para distinguirlo de las utilidades del propio sistema operativo y las aplicaciones del usuario. No obstante Unís viene equipado con una serie de servicios e interpretes que se consideran parte del sistema, estos pueden agruparse en una o varias Shell(interpretes de comandos), algún otro Sw de interfaz(un gestor de ventanas) y los componentes de un compilador. La capa exterior está formada por las aplicaciones del usuario.

Estos niveles son transparentes para el usuario y se puede decir que son los componentes internos del sistema operativo.

Externamente el sistema se manifiesta a través de una interfaz de usuario. En este aspecto, caben destacar dos tipos de sistemas operativos:

- Interfaz tipo texto. Se basa en una línea de comandos normalmente identificada por una letra(que está asociada a la unidad de almacenamiento masivo con la que el sistema trabaja por defecto), y/o por un símbolo (>,:,#,$) este identificador se denomina prompt del sistema, algunos de los sistemas operativos que utilizan este tipo de interfaz son: MS−DOS, LINUX−UNIX, CP/M, NOVELL NETWARE.
- Interfaz gráfica. Es una interfaz que está basada en una serie de ventanas y menús desplegables y normalmente contiene iconos(dibujos aclaratorios), estos iconos pueden proporcionar un acceso directo a funciones del sistema y a las aplicaciones del usuario, esta interfaz es más compleja en cuanto a su diseño, pero más difícil de interpretar para el usuario(Ej: Windows 98, OS/2).

Algunos sistemas operativos basados en una interfaz texto proporcionan una utilidad de ventanas que se ejecutan siempre como una utilidad más del sistema pero no como núcleo del sistema (X Windows, KDE, Gnome[en Linux], CDE[Unix], Windows 3.11[Dos]).

4.5. Modos de explotación.

Las distintas formas en las que se puede ejecutar un sistema y los procesos que controla dependen del número de usuarios que pretenden explotarlo, del número de procesos que se intentan ejecutar concurrentemente, del número de procesadores en la CPU y del tiempo de respuesta.

Dependiendo del número de usuarios que pueden explotar un sistema podemos distinguir;

- Sistema monousuario. Solo un usuario puede utilizar el ordenador en un tiempo determinado, todos los recursos del sistema se hallan disponibles para él.
- Sistema multiusuario. Cuando varios usuarios pueden ejecutarse simultáneamente procesos distintos y compartir determinados recursos (IBM−360). Dependiendo del número de procesos que se pueden ejecutar de forma simultanea podemos hablar de:
    - ♦ Monotarea o monoprogramación. Los procesos se ejecutan uno tras de otro secuencialmente, no se pasa el control de ejecución hasta que el anterior no acabe de ejecutarse completamente(explotación secuencial).
    - ♦ Multitarea o multiprogramación. Varios procesos están ejecutándose simultáneamente compartiendo todos tiempo de CPU, la asignación de tiempos de ejecución dependerá del sistema de explotación.

Dependiendo del número de procesadores que halla instalados en la CPU podemos hablar de:

- Sistema monoproceso. En los que solo hay un procesador.
- Sistema multiproceso. Es un CPU que consta de varios procesadores instalados, dedicados todos a ejecutar un mismo proceso o cada uno para un proceso diferente.

Según el tiempo que se tarde en obtener el resultado de un proceso ejecutado, un sistema informático puede trabajar en:

- Tiempo real. El tiempo de respuesta es inmediato a la ejecución del proceso.
- Tiempo compartido(timer sharing). Cada proceso va consumiendo un tiempo de CPU asignado previamente, cuando este tiempo es agotado se pasa el proceso de ejecución al siguiente.
- Por lotes(batch). Son procesos cuyo resultado no necesita ser procesado inmediatamente, en aquellos sistemas operativos que lo permitan como Linux−Unix y lo ejecutan en segundo plano.

5. Gestión del procesador: el núcleo, los procesos y la planificación. (Importante)

La mayoría de los procesadores dan soporte para al menos dos modos de ejecución, uno seria el modo usuario bajo el cual se ejecutarían las aplicaciones y el otro es el modo del sistema(núcleo o de control) bajo el cual se ejecutarían todos aquellos módulos vitales del sistema operativo.

Hay que tener en cuenta que ciertas interrupciones solo pueden ejecutarse bajo el modo de control como son por ejemplo aquellas que pertenecen a los módulos encargados de la gestión de la memoria o el control de archivos.

La razón por la que existen estos dos modos radica en la necesidad de proteger al sistema operativo y las tablas que genera el mismo para el control del sistema(tabla del bloque de memoria), de las ingerencias de las aplicaciones del usuario.

Cabe preguntarse como conoce el procesador en que modo ha de ejecutarse y cuando debe cambiar de modo. Para conocer el modo de ejecución en curso existe un bit que lo indica en el PSW(program status word) este bit cambia como respuesta a determinados sucesos, por ejemplo cuando un usuario hace una llamada a un servicio del sistema operativo o cuando una interrupción transfiere el control a una rutina del sistema operativo, se cambia el modo de ejecución al modo núcleo mediante una instrucción CHM(Change Mode).

Las funciones básicas del núcleo del sistema operativo son:

− Gestión de procesos.

- Creación y terminación de procesos.
- Planificación y expedición de procesos.
- Sincronización de procesos y soporte para la comunicación entre procesos.
- Gestión de los bloques de control de los procesos.

− Gestión de la memoria.

- Asignación de direcciones de memoria a los procesos.
- Gestión de paginas y segmentos.
- Gestión de swapping(ínter cambiabilidad memoria principal−disco)

− Gestión de la E/S.

- Gestión y control del buffering.
- Asignación de canales de E/S a los diferentes dispositivos así como los procesos.

− Funciones de soporte.

- Tratamiento de las interrupciones.
- Contabilidad.
- Supervisión.

5.1 Gestión de procesos. (Importante)

todos los sistemas operativos de multiprogramación, por ejemplo el Windows NT, hasta el MVS que da soporte a usuarios, pasando por Linux−Unix están construidos en torno al concepto de proceso. La misión principal del micro es ejecutar las instrucciones que se le dan, estas interrupciones están agrupadas secuencialmente en programas. Desde el punto de vista del procesador esta va ejecutando las instrucciones dentro de un repertorio, según la secuencia que viene dictada por los valores cambiantes de un registro.

El contador del programa, que no es más que un puntero que indica cual es la próxima instrucción que se ha de ejecutar. A lo largo del tiempo el valor del CP puede apuntar a instrucciones de diferentes programas que pertenecen a diferentes aplicaciones. La ejecución de un programa individual se llama proceso o tarea, el comportamiento de un proceso individual viene caracterizado por el listado de la secuencia de sus instrucciones, a este listado se le denomina traza del proceso.

Cualquier proceso cargado en memoria puede encontrarse en cualquier momento en alguno de los siguientes estados:

- Listo. Proceso que esta preparado para ejecutarse cuando se le de la oportunidad.
- Ejecución. El proceso está en curso.
- Bloqueado. El proceso que no puede ejecutarse hasta que no se termine un determinado proceso, por ejemplo la terminación de una operación de E/S.

Se puede decir que un proceso esta formado por los componentes siguientes:

- Un programa ejecutable.
- Los datos asociados necesarios para el programa(constantes, variables, espacios del trabajo, etc...).
- Contexto de ejecución del programa.

Este elemento es esencial, el contexto de ejecución incluye toda la información que el sistema operativo necesita para administrar el proceso y toda la información necesaria para que el micro pueda ejecutarlo correctamente.

Diferentes sistemas operativos organizan esta información de diferentes maneras, normalmente el contexto de ejecución se halla almacenado en una zona de proceso llamado bloque de control de procesos.

La información que puede usar el sistema operativo en este bloque sin detallar la forma en que la organización podía ser:

− Identificación de procesos.

- Identificadores. Los identificadores numéricos que se pueden guardar en un bloque de control de procesos se puede dividir en:
- Identificador de procesos.

- Identificador del proceso padre(si es que existe).
- Identificador del usuario.

− Identificador del estado del procesador o identificador del estado:

♦ Registros visibles para el usuario. Es aquel al que puede hacer referencia por medio de una instrucción maquina o por medio del ensamblador, normalmente existen de 8−32 de estos registros aunque hoy implementan RISC que disponen de 100 registros visibles.
♦ Registros de control y estado. Existen varios registros que se utilizan para controlar el funcionamiento del micro, entre ellos:

- El contador del programa.
- Códigos de condición(flags).
- Información de estado(incluye los indicadores de habilitación o inhabilitación de interrupciones y del modo de funcionamiento del micro).

♦ Punteros de pila. Cada proceso puede tener una o más pilas del sistema asociadas que se gestionan por el método Lifo. Las pilas sirven para gestionar las direcciones de retorno de los procedimientos y de las llamadas al sistema.

− Información de formación de procesos.

♦ Información de planificación y estado. Esta información la necesita el sistema operativo para llevar a cabo sus funciones a la planificación, contienen datos relativos a:

- Estado del proceso.
- Prioridad del proceso. Normalmente en multiprogramación cada proceso tiene un nivel de prioridad que es usado por el planificador para determinar el orden de ejecución y en la utilización de los recursos.
- Información sobre planificación. Dependiendo del algoritmo que utilice el planificador necesita un tipo de información u otra. Necesita saber cuanto tiempo lleva en esperar el proceso y que cantidad de tiempo de CPU gasto en su último turno de ejecución.
- Suceso; datos relativos a la naturaleza del proceso que esta esperando al proceso para ser reanudado.

♦ Estructuración de los datos. Un proceso puede estar formado con una estructura de cola, de anillo, de estrella, etc..., a demás un proceso puede estar enlazado con otro, en relación padre e hijo.

La información sobre este tipo de enlaces para poder gestionar este tipo de estructuras se halla en el bloque de control del proceso.

♦ Privilegios. A los procesos se les otorgan privilegios en términos de memoria a la que pueden acceder y los tipos de instrucciones que puede ejecutar, a demás puede haberlo respecto al uso de servicios y la utilización del sistema.
♦ Gestión de la memoria. Esta sección incluye punteros a las tablas de pagina y/o segmentos que se utilizan para la gestión de la memoria virtual.
♦ Propiedades de recursos y utilización. Se pueden indicar los recursos controlados por el proceso como por ejemplo los archivos abiertos también pueden incluir un histórico sobre la utilización del micro o de cualquier otro recurso.

Aunque los detalles pueden diferenciar de un sistema operativo a otro en general todos los sistemas organizan la información necesaria para la administración de los procesos en al menos cuatro categorías:

♦ Tablas de memoria. Se utilizan para registrar la estructura de la memoria interna y de la memoria virtual, parte de la memoria interna esta reservada al sistema operativo y el resto se utiliza para albergar los distintos procesos. Estos procesos se mantienen en parte en memoria auxiliar y se va cargando en memoria interna mediante un mecanismo de memoria virtual o mediante un sistema de ínter cambiabilidad memoria principal−disco(swapping).

Las tablas de memoria deben incluir información relativa a la asignación de direcciones de cada proceso, la asignación de memoria secundaria a los mismos, información relativa a que procesos pueden acceder a direcciones de memoria compartidas y cualquier otra información necesaria para gestionar la memoria virtual.

- Tabla de E/S. Son utilizadas para administrar los dispositivos y los canales de E/S, en un momento dado un dispositivo puede estar asignado a un proceso en concreto o disponible para otro. El sistema operativo debe conocer en todo momento esta situación y las direcciones de memoria que van a intercambiar en esta operación de E/S.
- Tabla de archivos. El sistema operativo almacena información referente a la dirección que ocupa en memoria secundaria, los tipos de acceso al archivo, las restricciones en su acceso y sus atributos.
- Tablas de procesos. Contienen información relativa a cada proceso como puede ser su identificación, su ubicación en memoria, el estado en el que se encuentra, la propiedad de los recursos que tiene, la prioridad, el usuario que lo ha lanzado, etc...

Cuando se añade un proceso a los que ya están en ejecución, este debe crear las estructuras de datos que se utilizan para administrarlo una o varias entradas en las tablas vistas anteriormente). A estas acciones se le denominan creación de un nuevo proceso.

En todo sistema también tiene que existir algún método para que el proceso pueda informar al sistema operativo de que ha acabado de ejecutarse. Normalmente esta situación se comunica mediante una sentencia tipo halt, utilizada mucho en procesos por lotes, también una acción del usuario puede dar lugar a que se termine el proceso. En cualquier caso este tipo de acciones provocan al final una petición de servicio al sistema operativo para que esta acabe con el proceso demandante a demás una serie de errores y fallos pueden llevar a la terminación del proceso en curso, estas situaciones están normalizadas según la tabla PINK 89.

Nota: Razones para la terminación de un proceso PINK 89(Pinker y Wear).

Situación Suceso que lo ha provocado

Terminación normal. El proceso ejecuta una llamada a un servicio del sistema operativo que indica que este ha terminado.

Terminación limite excedido. El proceso se ha ejecutado por más tiempo del limite total especificado.

No hay memoria disponible. El proceso necesita más memoria de la que el sistema operativo le puede proporcionar.

Violación de limites. El proceso ha intentado acceder a una posición de memoria que no le es permitida.

Error de protección. El proceso intenta acceder a un recurso o a un archivo para el cual no tiene permiso o lo esta utilizando de forma incorrecta.

Error aritmético. El proceso intenta realizar un proceso no permitido(Ej: dividir por 0) o intenta almacenar un dato que no cabe en el espacio almacenable del HW.

Tiempo máximo de espera rebasado.El proceso ha excedido el tiempo de espera para que haga un proceso determinado(Ej: que se le asigne una determinada impresora).

Instrucción no valida. El proceso ha intentado ejecutar una instrucción inexistente(Ej: un dato en vez

de una instrucción).

Instrucción privilegiada. El proceso ha intentado ejecutar una instrucción reservada a el sistema operativo.

Intervención del usuario o S.O. El operador o el sistema operativo acaban con el proceso.

Terminación del padre. Un sistema operativo puede estar diseñado para que cuando acabe un proceso padre se termine con todos sus hijos.

Solicitud del padre. Normalmente un proceso padre tiene el privilegio de acabar con cualquier de sus procesos hijos.

El sistema debe ser capaz de simultanear la ejecución de varios procesos de forma que

saque el mayor provecho posible al micro, ofreciendo a la vez un tiempo de respuesta

razonable, así mismo debe ofrecer y gestionar los recursos del sistema de acuerdo con

las prioridades de los procesos que lo solicitan evitando en todo momento el

Ínter bloqueo. El cual se produce cuando hay dos procesos que necesitan los mismos

recursos para continuar su ejecución y cada uno de ellos se ha apropiado de cada uno de

los recursos de forma que esperan indefinidamente a que el otro este disponible.

Un cambio de proceso puede producirse en cualquier momento en el que el sistema

operativo haya tomado el control a partir del proceso que esta ejecutando los sucesos

posibles que pueden dar el control al sistema operativo son:

- Una interrupción. Es un mecanismo generado por algún tipo de suceso que es externo e independiente del proceso que esta ejecutándose (Ej: La culminación de una operación de E/S).
- Un cepo. Tiene que ver con una condición de error o de excepción generada dentro del proceso que se esta ejecutando(Ej: El acceso ilegal a un archivo).
- Una llamada al supervisor. La daremos más adelante.

En una interrupción ordinaria el control se transfiere primero al gestor de interrupciones quien lleva a cabo unas tareas básicas y después pasa el control a la rutina del sistema operativo que se ocupa del tipo de interrupción que se ha producido.

Algunos ejemplos típicos de interrupción son:

- Interrupción de reloj. El sistema operativo determina si el proceso que esta ejecutando lo ha estado durante la fracción máxima de tiempo permitido, si este es así el proceso debe pasar a la situación de listo y el planificador debe asignar turno de ejecución al siguiente proceso.
- Interrupción de E/S. Cuando el sistema operativo determina que se ha producido una operación de E/S y esta constituye un suceso que han estado esperando una o más procesos, el sistema operativo debe pasar estos procesos del estado de bloqueado a listo, en ese momento deberá decidir el planificador si reanuda la ejecución del proceso que esta ejecutándose

actualmente o si lo expulsa en beneficio de cualquiera de los procesos en estado listo de mayor prioridad.

- Fallo en memoria. El procesador encuentra una referencia a una dirección de memoria virtual de una palabra que no se halla en memoria interna(fallo de pagina). El sistema operativo debe traer el bloque que contiene la palabra desde memoria secundaria a memoria interna. Cuando se produce el fallo en la referencia el proceso pasa a estado bloqueado y cuando el bloque es traído a memoria interna el proceso pasa a estado listo. En el caso de los cepos el sistema operativo determina si el error es o no fatal, en este caso el planificador da por terminado el proceso que lo causó y pasa el turno al siguiente proceso.

Sino es fatal dependiendo del diseño del sistema operativo se puede intentar ejecutar una rutina de recuperación del error, o simplemente informar de la naturaleza del mismo y seguir con la ejecución del proceso. Finalmente el sistema operativo puede activarse mediante una llamada al supervisor, por ejemplo esta ejecutándose un proceso de usuario y este solicita una operación de E/S como puede ser el acceso a un archivo, esta petición provoca la transferencia del control a una rutina que forma parte del código del sistema operativo(el gestor de archivo) y esta situación por lo general provoca que el proceso que ha causado la llamada pase a estado de bloqueado.

Pero ya dijimos que el sistema operativo funciona de la misma manera que cualquier otro Sw, es decir se trata de un programa ejecutado por el procesador, también sabemos que a menudo abandona el control y debe depender del micro para recuperarlo. Entonces si el sistema operativo es una colección de programas que son ejecutados por el procesador como cualquier otro Sw cabe preguntarse si se puede considerar al sistema operativo como un proceso, y en este caso quien lo controla a él.

Los diseñadores de sistemas operativos han dado respuestas a estas preguntas desde diferentes perspectivas(la respuesta es diferente para distintos sistemas operativos)

- Núcleo fuera de todo proceso.

Muchos de los sistemas operativos antiguos ejecutan el núcleo fuera de cualquier proceso, con este enfoque cuando el proceso en ejecución es interrumpido o hace una llamada al supervisor se salva el contexto de ejecución de este proceso y se pasa el control al núcleo del sistema operativo. Este tiene su propia región de memoria y su propio pila para gestionar las llamadas y los retornos a los procesos, posteriormente el sistema operativo puede restaurar el contexto de ejecución para reanudarlo, es decir se considera el concepto de proceso aplicado solo a los programas de usuario. El sistema operativo se ejecuta como una entidad separada que opera en modo privilegiado.

.....

- Ejecución dentro de los procesos de usuario.

Una alternativa común a los sistemas operativos de mini y micro−computadoras es ejecutar parte del Sw del sistema operativo en el contexto del proceso de usuario. El concepto se basa en que básicamente el sistema operativo es una colección de rutinas que el proceso de usuario va a llamar conforme las necesita, ejecutándose estas en el contexto del propio proceso. En un momento determinado el sistema operativo esta ejecutando a imágenes de procesos cada uno de los cuales incluye el contexto del proceso, su zona de proceso, su zona de datos y su correspondiente pila para gestionar los programas del núcleo.

.....

El código y los datos del sistema operativo están en el espacio de direcciones compartidas y por tanto son accesibles a todos los procesos. Cuando se produce una interrupción, un cepo o una llamada al supervisor el procesador cambia a modo núcleo y el control pasa al sistema operativo con tal fin se salva el contexto de ejecución del procesador y se cambia hacia el contexto de una rutina al sistema

operativo, sin embargo la ejecución continua en el contexto del mismo proceso de usuario, de esta manera no se ha llevado a cabo un cambio de proceso sino un cambio de contexto.

.....

− Sistemas operativos basados en procesos. Una ultima alternativa consiste en implementar el sistema operativo como una colección más de procesos del sistema, por supuesto y al igual que los casos anteriores el Sw de parte del núcleo se ejecutara en modo privilegiado, en este caso sin embargo algunas funciones del núcleo se organizan en procesos separados, no obstante debe haber una pequeña parte del código que quede fuera de todos los procesos incluidos los de las funciones del sistema operativo. Esta pequeña parte del código corresponde al planificador.

.....

Esta concepción del sistema operativo facilita el diseño modular que a su vez posibilita una fácil actualización por medio de nuevas rutinas y a demás posibilita una interfase clara y una comunicación sencilla entre los diferentes módulos.

Nota:

Micro−núcleos. Es un concepto que ha recibido mucha atención últimamente y se trata de un pequeño núcleo o micro−kernel que proporciona las bases a una ampliación modular, sin embargo el concepto esta muy difuso porque existen diferentes cuestiones que reciben distintas respuestas dependiendo del sistema operativo. Estas cuestiones hacen referencia a lo pequeño que debe ser el núcleo para ser tratado como tal, al diseño de los controladores o drivers para conseguir el mayor rendimiento a la vez que sus funciones se independizan del Hw, a si las operaciones que no sean del núcleo deben ejecutarse en el espacio del núcleo o en el del usuario y a si se debe mantener el código de los sub−sistemas existentes.

La filosofía en la que se basa el micro−núcleo es que solo las funciones absolutamente necesarias del sistema operativo deben permanecer en el núcleo, el resto se debe construir sobre el núcleo como subsistemas internos que interactúan con este(módulos de gestión de archivos, de seguridad, etc...).

5.2. Planificación del procesador.

El planificador de procesos es un elemento fundamental en los sistemas operativos multiprogramación−multiusuario, se encarga de indicar que proceso debe ejecutarse en cada momento y que proceso debe pasar de un estado a otro cuando se lee un determinado suceso como acabamos de ver.

Al conjunto de reglas y criterios en los que se basa el funcionamiento del planificador para tomar sus decisiones se denomina algoritmo de planificación. Entre las características que debe presentar un buen planificador y su algoritmo se puede citar:

- La imparcialidad. El planificador debe asignar el tiempo de utilización del microprocesador de la forma más equitativa y justa posible, sin perjudicar o beneficiar determinados procesos.
- La eficiencia. Debe intentar en todo momento ocupado al procesador, es decir debe evitar los tiempos muertos de CPU.
- Tiempo de respuesta. Debe minimizar el tiempo de respuesta para los procesos on−line que interactúan con el usuario.
- Rendimiento. Debe maximizar el número de trabajos que se ejecutan en un periodo de tiempo.

El principal problema con que se encuentra el planificador de procesos es que cada uno de estos es único e impredecible. Por ejemplo algunos procesos pasan mucho tiempo esperando operaciones de E/S y otros muy poco(aquellos cuya parte de código es de calculo). El planificador en principio no sabe cuanto tiempo pasara desde que un proceso pase del estado de ejecución al estado de listo, bloqueado o terminado.

Para determinar el tiempo lleva un proceso en ejecución activa una especie de cronometro que cuenta el tiempo de CPU consumido, el tiempo de CPU en turno de ejecución, el turno que lleva en estado de listo, etc... La información queda en la zona de datos que maneja el planificador para cada proceso(entradas en la tabla de procesos).

Existen varios algoritmos de planificación entre los que cabe destacar los siguientes:

- Tiempo paralelo o planificación por torneo. Consiste en crear una lista con los procesos pendientes de ser ejecutados, a cada uno de ellos se le asigna un periodo de tiempo de CPU (T) denominado cuantum. Cuando el proceso finaliza su cuantum el planificador pasa el control de ejecución al siguiente proceso hasta el final de la lista, para ello utiliza un registro contador activado por el reloj de la CPU, que provoca una interrupción cuando un proceso ha consumido el tiempo asignado.

Nota: Este sistema mejora en parte el antiguo algoritmo de planificación conocido como multiprogramación clásica, que consistía en cambiar de turno cuando el proceso en curso entraba en una E/S, esto provoca numerosos tiempos muertos de CPU cuando los procesos de ejecución poseían muchas operaciones de E/S.

Multiprogramación Clásica

E/S 1

E/S 2

E/S 3

E/S

P 3

E/S E/S

P 2

E/S

P 1

Tº muerto Tiempo de CPU

Tiempo Paralelo

E/S 3

E/S 2

E/S 1

E/S

P 3

P 2 E/S

P 1

Tiempo Muerto Tiempo de CPU

Este sistema también puede producir tiempos muertos de CPU, bien porque mediante el transcurso del cuantum el proceso entra en E/S, o bien porque cuando tiene turno aun se encuentra en estado de bloqueado esperando que finalice su operación de E/S.

- Time Sharing. Para solucionar los problemas de la multiprogramación en paralelo encontramos el algoritmo en tiempo que aprovecha estos tiempos muertos de CPU cambiando de turno de ejecución de un proceso cuando este agota su cuantum o cuando el proceso entra en E/S.

E/S 3

E/S 2

E/S 1

P 3

E/S

P 2

P 1

Cuantum Tiempo de CPU

- Lista de espera con intervalos múltiples,. Este algoritmo beneficia a los procesos que necesitan tiempo de CPU. Para ello asigna intervalos de tiempo de CPU,

Intentando premiar a aquellos procesos que permanecen durante más tiempo

sin pasar al estado de bloqueado, en principio todos los procesos tienen asignado

un cuantum que se incrementa en un 100% en el siguiente turno de ejecución. Si

el proceso en curso no pasa a estado bloqueado, por ejemplo por iniciar una

operación de E/S, obviamente el algoritmo del planificador debe establecer un

número limitado de cuantum en un solo turno de ejecución para evitar que un

proceso con escasas E/S monopolice al microprocesador indefinidamente al

resto de los procesos.

E/S 3

E/S 2

E/S 1

E/S

P 3

P 2

E/S

P 1

Tiempo de CPU

Cuantum

6. Gestión de la memoria.

En un sistema monoprogramado la memoria se divide en dos partes, una para alojar el núcleo del sistema operativo, y el otro para alojar los programas del usuario.

En un sistema multiprogramación esta parte destinada a las aplicaciones del usuario debe subdividirse en distintas zonas para poder albergar a los distintos procesos que se ejecutan de forma concurrente. Esta labor de subdivisión y administración de esta parte de la memoria la lleva a cabo el modulo gestor de memoria del sistema operativo.

Se entiende que una gestión eficaz de la memoria es esencial en un sistema multiprogramado, si solo hay unos pocos procesos ejecutándose a la vez probablemente pasaron mucho tiempo en estado de bloqueado esperando concluir sus operaciones de E/S lo que provocará numerosos tiempos muertos de CPU por ello es necesario repartir eficientemente la memoria disponible para poder albergar el número máximo posible de procesos que minimizan estos tiempos muertos de CPU.

6.1. Requisitos de la gestión de la memoria.

Los requisitos que se intentan satisfacer son normalmente cinco, según la normalización LIST 88 y son:

- Reubicación. Es obvio que el programador o conoce de antemano que programas van a ejecutarse concurrentemente con el suyo, ni que espacio van a ocupar. El sistema operativo debe ser capaz de poder cargar y descargar los procesos activos en la memoria principal para maximizar el uso del procesador y de mantener una gran cantidad de procesos en espera de poder ser ejecutados. Esto se consigue mediante técnicas de intercambiabilidad memoria principal−disco shaping, pero para poder llevar esto a cabo el sistema operativo debe saber de antemano donde coloca a los diferentes procesos y debe poder trasladar un proceso de una zona de la memoria a otro si el mecanismo del shaping así lo requiere. La información indispensable para llevar a cabo estas funciones es conocer la ubicación de la información de

control del proceso(tabla procesos), la pila de ejecución y la dirección base del proceso, a demás el Sw del sistema operativo y el Hw deben ser capaces de traducir las referencias a memoria encontradas en un programa a las direcciones físicas en memoria que reflejen la ubicación real de la referencia en un momento determinado.

- Protección. Cada proceso debe protegerse de interferencias no deseadas de otros procesos, ya sean intencionadas o no. Así pues el código de un determinado proceso no puede hacer referencia a direcciones de otros procesos(ya sea para leer o para escribir sin permiso). Sin duda la posibilidad de reubicar los procesos dificulta la protección de los mismos, ya que se desconoce la ubicación de un proceso en memoria durante la compilación para poder asegurar la protección de estas direcciones, esto obliga a que se deban hacer las comprobaciones en tiempo de ejecución, para poder asegurar que un proceso hace referencia exclusiva a direcciones de memoria de su zona, no obstante este tipo de protección debe hacerse mediante mecanismos Hw en vez de utilizar Sw, de echo es así como se hace porque ni siquiera el sistema operativo es capaz de anticiparse a las referencias en memoria de un programa, y en caso de poder hacerlo significaría tal consumo de CPU que colapsaría el sistema.
- Compartición. Cualquier mecanismo de protección debe ser lo suficientemente flexible como para permitir que los distintos procesos puedan acceder a direcciones de memoria compartidas, por ejemplo si varios procesos están ejecutando el mismo programa seria beneficioso para el sistema poder permitir a estos procesos acceder a una misma copia del mismo en vez de existir una copia para cada proceso.
- Organización física. En el esquema de organización de la memoria en dos niveles uno memoria interna más pequeña y volátil, y dos memoria auxiliar más grande y no volátil, es de suma importancia el control del flujo de información entre ellos. En principio podría pensarse que es el programador el que debe encargarse de controlar este flujo pero esto es impracticable e indeseable por:

- La memoria interna para albergar los procesos en sus datos puede ser insuficiente en un momento determinado.
- En el momento de la programación en un entorno multiprogramado no se conoce que procesos van a compartir memoria con uno determinado y que espacio van a ocupar y donde se va a encontrar este espacio.

Resulta claro pues que esta tarea de mover información de la memoria interna hasta la unidad de disco y viceversa debe ser responsabilidad del sistema, y de echo constituye la esencia del módulo gestor de memoria del sistema operativo.

- Organización lógica. La mayoría de los programas están organizados en módulos, si el sistema operativo y el Hw pueden tratar de forma efectiva los programas del usuario y sus módulos se podrían conseguir una serie de ventajas:
- Los distintos módulos pueden escribirse y compilarse de forma diferente, siendo el sistema el que resuelve en tiempo de ejecución las referencias de unos módulos a otros.
- Con un coste adicional bajo se puede hacer que los distintos módulos y aplicaciones posean distintos grados de protección(solo escritura, solo lectura, lectura−escritura).
- Es posible introducir mecanismos para la compartición de módulos.

Nota: El tipo de organización de la memoria que resuelve de forma más fácil estas necesidades es la segmentación.

Cuando se da la orden de ejecución de un programa este es cargado en memoria interna por el sistema operativo, quien le asigna una dirección base a partir de la cual se va a cargar y traducir las direcciones lógicas en direcciones físicas reales. La asignación de memoria a distintos programas que se van a ejecutar concurrentemente se realiza por algunos de los sistemas que vamos a ver a continuación o combinación entre ellos.

6.2. Particiones estáticas.

La memoria se divide en cierto número de zonas o particiones, cada una de las cuales va a albergar un programa. Las direcciones base que va a asignar el sistema operativo a cada programa van a ser las direcciones de comienzo de cada partición, el tamaño de cada partición puede ser determinado por el programador o por el sistema operativo(mayormente).

Los tamaños usuales son: 128 Kb, 256 Kb, 512Kb, o un Mb.

El sistema operativo mantiene una tabla en la que cada entrada(fila) corresponde a una partición, esta tabla posee información relativa a la dirección de comienzo de la partición, el tamaño(no todas tienen que ser del mismo tamaño), y el estado de la partición(ocupada o libre).

Cuando una partición quede libre el planificador hace que en ella se cargue el proceso de máxima prioridad que queda en ella, este método es ampliamente superado ya que presenta dos problemas esenciales.

- Puede que un programa no quepa en una partición lo que obliga al programador siempre a modularizar al máximo los programas.
- Efectúa un uso poco eficaz de la memoria, ya que cualquier programa por pequeño que sea va a ocupar una partición.

128 Kb

256 Kb

Zona de la partición

128 Kb Sin utilizar

256 Kb

6.3. Particiones dinámicas.

Las particiones son variables en número y longitud. Cuando se da la orden de ejecución de un proceso a este se le asigna la memoria que necesita exactamente. El sistema operativo gestiona una tabla de trabajos en la que cada fila contiene información acerca del identificador del trabajo, el espacio de memoria que ocupa y la dirección base de comienzo de cada partición, también se mantiene de forma paralela una tabla con información relativa a los huecos libres y el espacio que ocupan, para asignar memoria a los programas el planificador consulta periódicamente el contenido de las tablas.

como consecuencia de la terminación de algunos programas van quedando huecos libres en memoria esto puede hacer que en un momento determinado se pueda aprovechar muy poca memoria en el sistema. Este problema puede solucionarse aglutinando los diferentes fragmentos libres aprovechando la característica de reubicación de los procesos. Esto es lo que se conoce con el nombre de compactación.

Situación inicial P3 Termina P1 Termina P6 quiere ejecutarse

Los tres métodos o algoritmos de ubicación que se pueden considerar son:

- ◊ El de mejor ajuste o best−fit. El módulo gestor de memoria elige la partición de tamaño más parecido al del proceso que solicita la expresión.
- ◊ El primer ajuste o first−fit. El gestor recorre el mapa de memoria desde el principio y elige el primer hueco de memoria que sea lo suficientemente grande.
- ◊ Siguiente ajuste o next−fit. Recorre la memoria a partir de la ultima ubicación eligiendo el primer bloque disponible que sea lo suficientemente grande.

6.4. Paginación.

Con este procedimiento la memoria se estructura en bloques de longitud fija que suele ser de 1,2, o 4 Kbyte. Cada uno de los cuales viene identificado por un número correlativo, así mismo los procesos se dividen en partes correlativas llamadas paginas, para un sistema dado el tamaño del bloque y la pagina coinciden.

Este sistema se fundamenta en el echo de que no es necesario que para que un proceso pueda ejecutarse no todo su código debe estar en memoria interna, basta con que este la pagina que contenga la instrucción en curso. En todo caso ni siquiera tiene la obligación de que todas las paginas de un proceso estén en bloques consecutivos en la memoria interna cuando el proceso valla a ejecutarse. En paginación las paginas se almacenan en bloques libres independientemente de si están contiguos o no, obviamente dentro de una misma pagina las instrucciones si ocupan posiciones consecutivas, de forma que una instrucción viene referenciada por la dirección base del bloque que ocupa más la posición relativa de la posición dentro de la pagina.

Para gestionar la paginación el sistema operativo mantiene tres tablas:

- ◊ Tabla−mapa de paginas. Existe una por cada programa y contiene información relativa al bloque en el que se encuentra cada uno de las paginas del mismo. Es de longitud variable y depende del número de paginas de cada programa.
- ◊ Tabla de bloques de memoria. Existe una entrada por cada uno de los bloques en los que se ha dividido la memoria, y almacena información relativa al identificativo del programa que ocupa cada bloque y en su caso si el bloque está libre o no.
- ◊ Tabla de procesos. Contiene una fila por cada uno de los trabajos que se hallan en ejecución. Contiene información a cerca del tamaño, dirección de memoria interna donde se encuentra su tabla−mapa de pagina, el estado y la situación del programa.

6.5. Segmentación.

El programa se considera dividido en segmentos, hay que tener en cuenta que en este caso cuando nos referimos a segmento lo estamos haciendo a un grupo lógico de información (un programa, un ,módulo, una pila, una sub−rutina, etc...). Por tanto se puede decir que un programa estará compuesto por diferentes segmentos que no tienen porque ser todos del mismo tamaño.La gestión la realiza el sistema operativo de forma análoga a como lo hace en el caso de las particiones dinámicas, solo que en vez de albergar cada partición a un programa ahora albergará un segmento del programa. El sistema contiene una tabla−mapa de segmentos parecida a la de la paginación, la segmentación se puede realizar de forma similar o en combinación con esta ultima, ambas técnicas permiten que ciertos programas o zonas de memoria puedan ser usadas simultáneamente por distintos procesos(compartición). Por tanto en un momento dado existirán en memoria bloques ocupados, bloques compartidos y bloques libres. El estado del bloque se recoge en una entrada de la tabla de bloques de memoria.

La diferencia esencial con una partición dinámica es que en segmentación un programa puede ocupar más de una partición y además los segmentos de un mismo programa no tienen porque estar consecutivos en memoria interna.

6.6. Memoria Virtual.

La memoria virtual es una técnica que permite el uso de un programa cuyo tamaño excede a la capacidad que físicamente tiene la memoria de la computadora, esta técnica se resuelve bajo dos enfoques principalmente, la paginación y la segmentación, aunque se puede utilizar la combinación de ambos. Para gestionar la memoria virtual necesitamos soporte Hw y Sw, el soporte Hw lo proporciona el microprocesador del sistema, que es el que se encarga de traducir las direcciones de memoria virtuales en direcciones físicas reales, además de las interrupciones provocadas cuando una pagina o segmento hace referencia a una instrucción que no se halla en memoria interna(se genera un fallo de pagina).

Esta interrupciones generadas por el microprocesador activan el Sw del módulo gestor de memoria del sistema operativo.

Considerando el aspecto Hw, la memoria virtual independientemente de si se basa en paginación, segmentación o combinación presenta estos dos aspectos:

- ◊ Todas las referencias a memoria dentro de un proceso se hace por medio de direcciones lógicas que se traducen dinámicamente a direcciones físicas reales por mecanismos Hw(en tiempo de ejecución).
- ◊ Un proceso puede dividirse en partes ya sean paginas o segmentos y no es necesario que estas partes se encuentren consecutivas en memoria interna durante la ejecución.
- ◊ La consecuencia inmediata de estas dos ventajas es que no resulta imprescindible que todas las paginas o segmentos de un programa se encuentren en memoria interna para poder ser ejecutados, basta con que se encuentren las paginas o segmentos que contengan las instrucciones en curso o como mucho las que se prevean que se van a ejecutar en un futuro inmediato. La división se aplica también a la zona de datos y no solo a los programas.

Todo esto supone unas mejoras en la utilidad del sistema puesto que:

- ♦ Se pueden conservar más procesos en memoria ya que hay más espacio disponible y esto redunda en una minimización de los tiempos muertos de CPU(disminuye la probabilidad de que en un momento dado todo los proceso estén en E/S).
- ♦ Es posible ejecutar procesos muy grandes e incluso un solo proceso puede exceder ampliamente la capacidad de la memoria interna, ya que parta se ejecute solo se necesita que este cargada la pagina o segmento que tenga la instrucción en curso, mientras que el resto permanece en la memoria auxiliar(disco).

Como los procesos solo se pueden ejecutar cuando están en la memoria interna a esta memoria se le llama memoria real pero un programador o un usuario percibe en potencia una memoria mucho mayor que la que utiliza el disco como soporte, a esta memoria se le denomina memoria virtual.

Con este sistema e un momento dado solo esta en memoria interna unos pocos segmentos de los procesos de ejecución. Esto va a hacer que en ocasiones tengamos que evacuar fragmentos de memoria interna para dar cabida a otros que contienen instrucciones que deben ser ejecutadas. Además si se expulsa un fragmento que contiene instrucciones o datos que van a ser referenciados en un futuro inmediato este deberá ser encargado de nuevo rápidamente.

En definitiva se están evacuando continuamente fragmentos desde la memoria interna, lo que puede suponer un handicap que dificulte el uso de esta técnica, puesto que puede conducir a los que se conocen como trashing o hiperpaginación, es decir que el microprocesador

consume más tiempo en las operaciones de shaping que ejecutando instrucciones.

Las formas de evitar el trashing constituyeron una de las áreas más importantes de investigación en los años 70, llegándose a una serie de algoritmos complejos pero que solucionaban el problema, en esencia el sistema operativo intenta deducir basándose en el historia reciente de cada proceso que fragmentos se usaran con menor probabilidad en un futuro próximo, basándose en el principio de cercanía y en que los programas están redactados de forma lineal(no abundan los saltos entre posiciones de memoria muy alejadas).

De todo esto se deduce que durante periodos cortos de tiempo se necesitan unos solos fragmentos de los procesos en curso, base en la que se sustenta también la predicción sobre que fragmentos se van a necesitar en un futuro próximo para poder evitar la hiperpaginación.

La gestión de la memoria virtual con demanda de paginación combina las técnicas de paginación e intercambiabilidad o shaping por lo general se utiliza un método de intercambio llamado lazy swapper o intercambio perezoso, por el cual solo se lleva a memoria interna las paginas en ejecución de los programas en procedimiento concurrente, la gestión la realiza el sistema operativo ayudándose de las siguientes tablas:

- ◊ Tabla−plano de trabajo. El sistema operativo crea y mantiene una de estas tablas por cada proceso, cada fila de la tabla corresponde a una de las paginas del proceso y contiene al menos los siguientes campos.

- ♦ Dirección de memoria principal, donde se encuentra la dirección de la pagina en disco.
- ♦ Bloque de memoria principal, en el que en su caso se encuentra dicha pagina.
- ♦ Un bit de estado, que indica la situación de la pagina(0en memoria interna, y 1 en memoria auxiliar).

- ◊ Tabla de bloques de memoria. Cada fila de esta tabla corresponde a un bloque de memoria y corresponde a la siguiente información.

- ♦ Identificativo del programa y de la pagina del programa que esta en ese bloque.
- ♦ Clave de protección. A cada programa se le asigna una clave y no se puede acceder a una pagina salvo que coincida la clave de la pagina y la del bloque.
- ♦ Estado de bloque(0 ocupado, y 1 libre).
- ♦ Bit de cambio. Se utiliza para indicar si desde que se cargo la pagina del disco por ultima vez se ha modificado esta o no(1 y 0, respectivamente).
- ♦ Contador de frecuencia. Que indica el número de veces que se ha utilizado la paginación desde la ultima vez que se introdujo en memoria interna una nueva pagina.

Cuando un proceso hace referencia a una pagina que no esta copiada en memoria interna se dice que se ha producido un fallo, el sistema operativo busca en la memoria interna una tabla de bloques de memoria un bloque que este libre, si existe alguno extrae de la tabla plano de trabajo la dirección en memoria principal donde se halla la dirección de la pagina en el disco que tiene que cargar, después de cargarla actualiza de nuevo todas las tablas. En el caso de que no haya ningún bloque libre se busca el menos referenciado(consultando al contador de frecuencia) y si su bit de cambio es 0, es decir la pagina no ha sido modificada y es copia exacta de la que hay en el disco, se graba sobre él sin más la nueva pagina, si por el contrario el bit de cambio es 1, antes de cargar la nueva pagina se actualiza en disco la pagina que se ha de evacuar, a partir de la dirección que se obtiene consultando la tabla plano de trabajos.

6.6.1. La gestión de la memoria en Windows NT.

Windows NT fue diseñado para trabajar en diferentes plataformas, una de las más importantes es la familia Intel X−86, en este caso adopta un tamaño de pagina de 4 Kbytes, que es la base

en la que se monto el esquema de la memoria virtual a partir del Intel 80486, Intel ya había incorporado mecanismos Hw tanto para la segmentación como para la paginación en el modelo Intel 80386, ambos mecanismos pueden utilizarse con lo que podemos escoger entre cuatro modelos distintos de gestionar la memoria.

◊ Memoria no paginada, ni segmentada.

En este caso coincide la dirección virtual con la dirección física real, lo cual es útil para procesos de control de poca complejidad y alto rendimiento.

◊ Memoria paginada y no segmentada.

La memoria se ve como un espacio lineal de direcciones paginadas. El algoritmo de pagina se encarga de la protección y gestión de la memoria virtual. Este sistema es el que eligen sistemas operativos como el UNIX.

◊ Memoria segmentada y no paginada.

La memoria se contempla como un espacio lineal de subconjuntos de direcciones lógicas. Las ventajas de este modelo respecto a la paginación es que permite protección a nivel de byte y además garantiza que la tabla de segmentos estará accesible al microprocesador cuando el segmento este en memoria, esto implica que los tiempos de acceso sean predecibles.

◊ Memoria segmentada y paginada.

La memoria se utiliza para definir particiones lógicas de memoria sometidas a un control de accesibilidad por parte del módulo gestor de memoria y la paginación se usa para gestionar la asignación de memoria dentro de cada uno de estas particiones. Otros sistemas operativos como OS−DOS se han decantado por este modelo.

6.6.1.1 Segmentación.

Cuando se emplea segmentación cada dirección virtual lógica esta formada por:

◊ 16 Bits(0−15) para identificar el segmento de los cuales los 2 primeros sirven para implementar el mecanismo de protección.
◊ 32 Bits para el desplazamiento.

Dirección Virtual

0 15 / 0 31

Así pues con memoria virtual n segmentada la memoria virtual del usuario es de 232 bits, igual a 4Gbytes posibles. Con la segmentación podemos tener direcciones de 246, igual a 64 Tbytes como longitud del espacio total visible para el usuario(obviamente para direccionar la memoria física real solo se puede utilizar los 32 bit de la longitud de palabra de los Intel).

Asociado a cada segmento hay dos formas de protección, el nivel de privilegio y atributo de acceso.

Existen 4 niveles de protección desde el más alto o nivel 0 al más bajo o nivel 3.

El nivel de privilegio asociado a un segmento que contiene datos se denomina clasificación, el nivel de privilegio asociado a un segmento de programa se denomina acreditación.

Un segmento de programa tiene acceso a un segmento de datos si su acreditación es menor o igual que la clasificación de este ultimo(es decir tiene mayor privilegio).

El Hw no dicta como se han de usar estos niveles de privilegio sino que su uso depende del diseño y la implementación del sistema operativo, en Windows NT se pretendió que el sistema operativo usara en su mayor parte el nivel de privilegio 1, dejando el nivel 0 para una pequeña parte del mismo encargada de la gestión, protección y control de acceso a la memoria, y los niveles 2 y 3 se reservaron para las aplicaciones. En muchos sistemas las aplicaciones tiene nivel 3 y el nivel 2 no se usa. En otro el nivel 2 lo poseen aquellas aplicaciones que deben estar protegidas debido a que implementan sus propios mecanismos de seguridad como por ejemplo los motores−gestores de base de datos, los entornos de ingeniería del Sw o los sistemas de automatización.

Además de regular el acceso a los segmentos de datos los mecanismos de privilegios limitan el uso de cierta instrucciones, como por ejemplo aquellas que se ocupan de la gestión de los registros de la memoria que solo pueden ejecutar en nivel 0, o las instrucciones de E/S que normalmente solo se pueden ejecutar hasta un nivel asignado por el sistema operativo y que suele ser el nivel 1.

El atributo de acceso de un segmento de datos determina si se permiten accesos de solo lectura o de lectura / escritura, en el caso de segmentación de programas este atributo determina lectura o lectura / ejecución. El mecanismo de traducción de direcciones en la segmentación implica la transformación de una dirección virtual en lo que se denomina una dirección lineal. El formato de la dirección virtual incluye información relativa a:

◊ Indicación de tablas.

Indica si en la traducción debe utilizar la tabla de segmentos global o la tabla de segmentos local. La mitad del espacio virtual de direcciones(8 K segmentos por 4Gbytes) se considera global, es decir compartida por todos los segmentos, el resto de la memoria se considera local y distinta para cada segmento.

◊ Número de segmentos.

Que sirve como índice para acceder a la tabla de segmentos.

◊ Desplazamiento.

Es la posición relativa del byte dirección dentro del segmento.

◊ RPL.

Request Privilege Level(nivel de privilegio solicitado).

El formato de la dirección lineal es.....(Ver fotocopia).

Cada entrada en la tabla de segmentos consta de 64 bits y tiene los siguientes datos:

- Limite. Define el tamaño del segmento. Dependiendo del bit de granulidad esta magnitud se interpreta en unidades de un Byte(hasta un máximo de un Mega) o unidades de 4 Kbytes(hasta un Gbyte).
- Dirección base. Define la dirección de comienzo del segmento dentro del

espacio lineal de 4Gbytes.

3) Bit de acceso. Se pone activo(1) si se ha accedido a ese segmento, también puede

utilizarse para controlar la frecuencia de utilización del segmento en cuestión.

4) Tipo. Especifica el tipo de segmento y se utiliza para la clasificación y la

acreditación.

5) Nivel de privilegio (0,1,2, o 3).

6) Bit de presencia. Indica si el segmento esta en memoria interna o no.

- Bit de granulidad. Indica si hay que interpretar el campo limite en unidades de un Byte o 4 Kbytes.

6.6.1.2. Paginación.

La segmentación es un servicio opcional y puede inhabilitarse, cuando se usan las direcciones del programa son direcciones virtuales que se convierten en direcciones físicas reales en tiempo de ejecución como acabamos de ver. Cuando no se usa el programa utiliza directamente direcciones lineales, en este caso el paso siguiente consiste en convertir estas direcciones lineales en direcciones físicas de 32 bits.

Para comprender las estructuras de las direcciones lineales hace falta saber que el mecanismo de paginación en la familia Intel X−86 es una operación de búsqueda en dos niveles a una tabla. El primer nivel consiste en la consulta de un directorio de paginas que pueden contener hasta 1024 entradas, este directorio divide el espacio lineal de memoria en 4 Gbytes en 1024 grupos de 4 Mbytes de longitud, cada uno de los cuales posee su propia tabla de paginas. Cada tabla contiene hasta 1024 entradas, cada una de las cuales corresponde a una pagina de 4 Kbytes(Grafico).

La gestión de memoria tiene posibilidades de utilizar un mismo directorio de pagina para todos los procesos, un directorio de pagina para cada proceso o una combinación de ambos. En todo caso el proceso del directorio de pagina del proceso en curso debe estar en memoria interna y el resto puede estar en ese mismo momento en memoria virtual.

C.O Op A Op B Op C

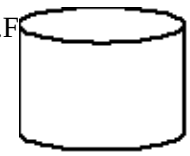
C.O Op A y Rdo OpB

C.O Dir n

Dir k

DATO

F.F



uente

F.Fuente

F.Fuente

Compilador

F.Objeto

P.Objeto

Interprete

Micro

Micro

F.Fuente

A. Léxico−Gráfico

A. Sintáctico

A. Semántico

Optimización

?

Generación del fichero objeto

F.Objeto

Tablas y árboles

A

B

C

E

D

Nodo hijo

Cod. Soc

USUARIO

PROGRAMAS APLICACIÓN

UTILIDADES

SISTEMA OPERATIVO

HARDWARE

NÚCLEO

P1

P2

Pn

P1

Funciones

S.O.

P2

Funciones

S.O.

Pn

Funciones

S.O.

P1

F. S.O.

Pn

F. S.O.

P2

F. S.O.

Funciones de control de procesos

P1

S.On

Pn

S.O2

P2

S.O1

Funciones de control de procesos

P1

P2

P3

P4

P4

Nº Segmento Desplazamiento

P2

P2

P4

P1

P2

P4

P1

P3

P4

P2