

SISTEMAS OPERATIVOS

EXPTO. DAVID LUIS
LA RED MARTINEZ

PROFESOR TITULAR ORDINARIO DE
COMPUTACION V (SISTEMAS OPERATIVOS)

LICENCIATURA EN SISTEMAS

SISTEMAS OPERATIVOS

- v INTRODUCCION
- v PROCESOS Y ADMINISTRACION DEL PROCESADOR
- v ADMINISTRACION DE LA MEMORIA
- v SISTEMAS DE ARCHIVOS
- v ENTRADA / SALIDA
- v BLOQUEOS
- v INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS
- v COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS
- v SINCRONIZACION EN SISTEMAS DISTRIBUIDOS
- v PROCESOS Y PROCESADORES EN SISTEMAS DISTRIBUIDOS
- v SISTEMAS DISTRIBUIDOS DE ARCHIVOS
- v RENDIMIENTO
- v MODELADO ANALITICO EN RELACION AL RENDIMIENTO
- v SEGURIDAD DE LOS SISTEMAS OPERATIVOS
- v ESTUDIO Y COMPARACION DE DISTINTOS SISTEMAS OPERATIVOS

INTRODUCCION

- v QUE ES UN SISTEMA OPERATIVO
- v HISTORIA DE LOS SISTEMAS OPERATIVOS - GENERACIONES
- v CONCEPTOS DE LOS SISTEMAS OPERATIVOS
- v ESTRUCTURA DE LOS SISTEMAS OPERATIVOS
- v TENDENCIAS
- v HARDWARE
- v SOFTWARE
- v MEMORIA FIJA

QUE ES UN SISTEMA OPERATIVO

- v ES UN GRUPO DE PROGRAMAS DE PROCESO CON LAS RUTINAS DE CONTROL NECESARIAS PARA MANTENER CONTINUAMENTE OPERATIVOS DICHS PROGRAMAS.
- v OBJETIVO PRIMARIO DE UN SISTEMA OPERATIVO:
 - u OPTIMIZAR TODOS LOS RECURSOS DEL SISTEMA PARA SOPORTAR LOS REQUERIMIENTOS.
- v CLASIFICACION DEL SOFTWARE PARA COMPUTADORAS:
 - u PROGRAMAS DE SISTEMA:
 - Φ CONTROLAN LA OPERACION DE LA COMPUTADORA EN SI.
 - u PROGRAMAS DE APLICACION:
 - Φ RESUELVEN PROBLEMAS PARA LOS USUARIOS.
- v EL SISTEMA OPERATIVO ES EL PROGRAMA FUNDAMENTAL DE TODOS LOS PROGRAMAS DE SISTEMA.
- v EL S. O. PROTEGE Y LIBERA A LOS PROGRAMADORES DE LA COMPLEJIDAD DEL HARDWARE:
 - u SE COLOCA UN NIVEL DE SOFTWARE POR SOBRE EL HARDWARE PARA:
 - Φ CONTROLAR TODAS LAS PARTES DEL SISTEMA.
 - Φ PRESENTAR AL USUARIO UNA INTERFAZ O MAQUINA VIRTUAL.
- v ESQUEMA TIPICO DE UN SISTEMA DE COMPUTOS:
 - u PROGRAMAS DE APLICACION:
 - Φ SISTEMA BANCARIO, RESERVACIONES EN UNA LINEA AEREA, JUEGOS, ETC..
 - u PROGRAMAS DE SISTEMA:
 - Φ COMPILADORES, EDITORES, INTERPRETES DE COMANDOS.
 - Φ SISTEMA OPERATIVO.
 - u HARDWARE:
 - Φ LENGUAJE DE MAQUINA.
 - Φ MICROPROGRAMACION.
 - Φ DISPOSITIVOS FISICOS.

QUE ES UN SISTEMA OPERATIVO

v **MICROPROGRAMA:**

- ▮ SOFTWARE QUE GENERALMENTE SE LOCALIZA EN LA MEMORIA DE SOLO LECTURA.
- ▮ BUSCA LAS INSTRUCCIONES DE LENGUAJE DE MAQUINA PARA EJECUTARLAS COMO UNA SERIE DE PEQUEÑOS PASOS.
- ▮ EL CONJUNTO DE INSTRUCCIONES QUE INTERPRETA DEFINE AL **LENGUAJE DE MAQUINA**.
- ▮ EN CIERTAS MAQUINAS SE IMPLANTA EN EL HARDWARE Y NO ES EN REALIDAD UNA CAPA DISTINTA.

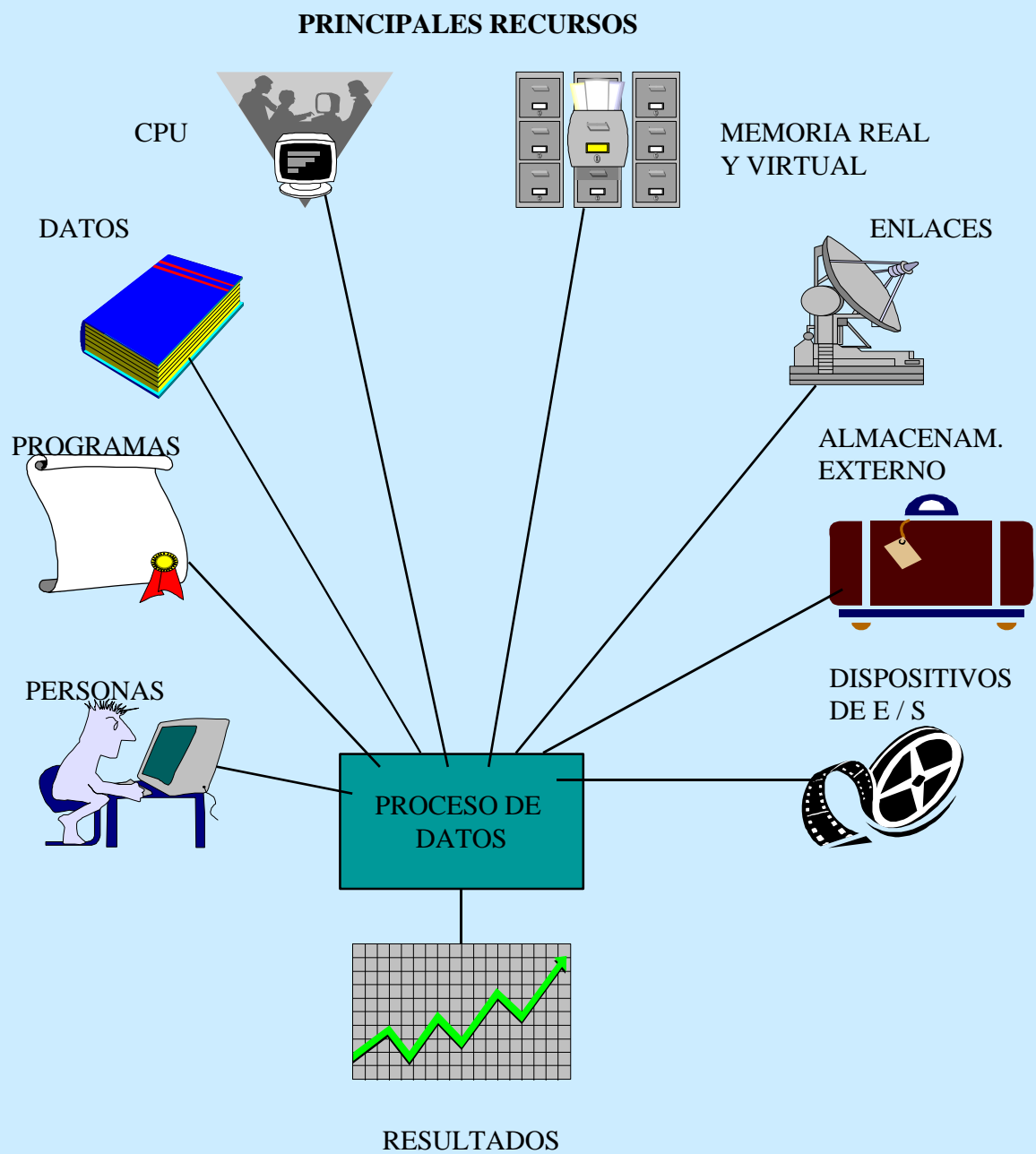
v **LENGUAJE DE MAQUINA:**

- ▮ GENERALMENTE POSEE ENTRE 50 Y 300 INSTRUCCIONES, SIRVIENDO LA MAYORIA PARA DESPLAZAR DATOS, HACER OPERACIONES ARITMETICAS Y COMPARAR VALORES.
- ▮ LOS DISPOSITIVOS DE E/S SE CONTROLAN AL CARGAR VALORES EN REGISTROS DEL DISPOSITIVO ESPECIALES.
- v UNA DE LAS PRINCIPALES FUNCIONES DEL S. O. ES OCULTAR TODA ESTA COMPLEJIDAD Y BRINDAR AL PROGRAMADOR UN CONJUNTO MAS CONVENIENTE DE INSTRUCCIONES PARA TRABAJAR.
- v EL S. O. SE EJECUTA EN MODO CENTRAL O MODO DE SUPERVISION, CON MAXIMA PRIORIDAD Y GENERALMENTE CON PROTECCION POR HARDWARE.
- v LOS COMPILADORES, EDITORES Y DEMAS PROGRAMAS SE EJECUTAN EN MODO USUARIO.
- v EL S. O. ES LA SERIE DE PROGRAMAS, DISPUESTOS YA SEA EN EL SOFTWARE O EN LA MEMORIA FIJA (MICROCODIGO), QUE HACEN AL HARDWARE UTILIZABLE.
- v LOS S. O. PONEN EL “PODER COMPUTACIONAL BASICO” DEL HARDWARE CONVENIENTEMENTE A DISPOSICION DEL USUARIO, PERO CONSUMEN PARTE DE ESE PODER COMPUTACIONAL PARA FUNCIONAR.
- v LOS S. O. SON, EN PRIMER LUGAR, ADMINISTRADORES DE RECURSOS, SIENDO EL RECURSO PRIMARIO EL HARDWARE DEL SISTEMA.

QUE ES UN SISTEMA OPERATIVO

- v PRINCIPALES CARACTERISTICAS DE LOS S. O.:
 - u DEFINIR LA “INTERFAZ DEL USUARIO”.
 - u COMPARTIR EL HARDWARE ENTRE USUARIOS.
 - u PERMITIR A LOS USUARIOS COMPARTIR LOS DATOS ENTRE ELLOS.
 - u PLANIFICAR RECURSOS ENTRE USUARIOS.
 - u FACILITAR LA ENTRADA/SALIDA.
 - u RECUPERARSE DE LOS ERRORES.
- v PRINCIPALES RECURSOS ADMINISTRADOS POR LOS S. O.:
 - u PROCESADORES.
 - u ALMACENAMIENTO.
 - u DISPOSITIVOS DE E/S.
 - u DATOS.
- v LOS S. O. SON UNA INTERFAZ CON:
 - u OPERADORES.
 - u PROGRAMADORES DE APLICACIONES.
 - u PROGRAMADORES DE SISTEMAS (ADMINISTRADORES DEL S. O.).
 - u PROGRAMAS.
 - u HARDWARE.
 - u USUARIOS.
- v EL S. O. DEBE PRESENTAR AL USUARIO EL EQUIVALENTE DE UNA **MAQUINA EXTENDIDA** O **MAQUINA VIRTUAL** QUE SEA MAS FACIL DE PROGRAMAR QUE EL HARDWARE SUBYACENTE.

QUE ES UN SISTEMA OPERATIVO



HISTORIA DE LOS S. O. - GENERACIONES

- v LOS S. O. HAN ESTADO RELACIONADOS HISTORICAMENTE CON LA ARQUITECTURA DE LAS COMPUTADORAS EN LAS CUALES SE EJECUTAN.
- v GENERACION CERO (DECADA DE 1940):
 - u CARENCIA TOTAL DE S. O.
 - u COMPLETO ACCESO AL LENGUAJE DE MAQUINA.
- v PRIMERA GENERACION (1945-1955): BULBOS Y CONEXIONES:
 - u CARENCIA DE S. O.
 - u EN LOS AÑOS CINCUENTA COMIENZAN COMO TRANSICION ENTRE TRABAJOS, HACIENDO LA MISMA MAS SIMPLE.
- v SEGUNDA GENERACION (1955-1965): TRANSISTORES Y SISTEMAS DE PROCESAMIENTO POR LOTES (BATCH):
 - u EN LOS AÑOS SESENTA APARECEN LOS S. O. PARA SISTEMAS COMPARTIDOS CON:
 - Φ **MULTIPROGRAMACION:** VARIOS PROGRAMAS DE USUARIOS SE ENCUENTRAN AL MISMO TIEMPO EN EL ALMACENAMIENTO PRINCIPAL, CAMBIANDO EL PROCESADOR RAPIDAMENTE DE UN TRABAJO A OTRO.
 - Φ **MULTIPROCESAMIENTO:** VARIOS PROCESADORES SE UTILIZAN EN UN MISMO SISTEMA PARA INCREMENTAR EL PODER DE PROCESAMIENTO.
 - u POSTERIORMENTE APARECE LA **INDEPENDENCIA DE DISPOSITIVO:**
 - Φ EL PROGRAMA DEL USUARIO ESPECIFICA LAS CARACTERISTICAS DE LOS DISPOSITIVOS QUE REQUIEREN LOS ARCHIVOS.
 - Φ EL S. O. ASIGNA LOS DISPOSITIVOS CORRESPONDIENTES SEGUN LOS REQUERIMIENTOS Y LAS DISPONIBILIDADES.
- v TERCERA GENERACION (1965-1980): CIRCUITOS INTEGRADOS Y MULTIPROGRAMACION:
 - u DIFUSION DE LA **MULTIPROGRAMACION:**
 - Φ PARTICION DE LA MEMORIA EN PORCIONES, CON TRABAJOS DISTINTOS EN C/U DE ELLAS.

HISTORIA DE LOS S. O. - GENERACIONES

- ❖ APROVECHAMIENTO DEL TIEMPO DE ESPERA CONSECUCIA DE OPERACIONES DE E/S, PARA UTILIZAR LA CPU PARA OTROS PROCESOS.
- υ PROTECCION POR HARDWARE DEL CONTENIDO DE CADA PARTICION DE MEMORIA.
- υ APARICION DE TECNICAS DE **SPOOLING**:
 - ❖ **SIMULTANEOUS PERIPHERAL OPERATION ON LINE**: OPERACION SIMULTANEA Y EN LINEA DE PERIFERICOS.
 - ❖ **ALMACENAMIENTO DE TRABAJOS DE ENTRADA Y DE SALIDA EN DISPOSITIVOS TRANSITORIOS RAPIDOS (DISCOS)**, PARA DISMINUIR EL IMPACTO DE LOS PERIFERICOS MAS LENTOS.
- υ SON **SISTEMAS DE MODOS MULTIPLES**: DEBEN SOPORTAR **SISTEMAS DE PROPOSITOS GENERALES**: SON GRANDES Y COMPLEJOS PERO MUY PODEROSOS.
- υ INTERPONEN UNA **CAPA DE SOFTWARE** ENTRE EL USUARIO Y EL HARDWARE.
- υ APARECEN LOS **LENGUAJES DE CONTROL DE TRABAJOS**: NECESARIOS PARA ESPECIFICAR EL TRABAJO Y LOS RECURSOS REQUERIDOS.
- υ SOPORTAN **TIMESHARING (TIEMPO COMPARTIDO)**: VARIANTE DE LA MULTIPROGRAMACION CON USUARIOS CONECTADOS MEDIANTE **TERMINALES EN LINEA**:
 - ❖ OPERACION EN **MODO INTERACTIVO O CONVERSACIONAL**.
- υ APARECEN LOS **SISTEMAS DE TIEMPO REAL**: REQUIEREN TIEMPOS DE RESPUESTA MUY EXIGENTES, ESPECIALMENTE PARA USOS INDUSTRIALES O MILITARES.
- υ SE DIFUNDEN LAS COMPUTADORAS DE RANGO MEDIO.
- υ CUARTA GENERACION (1980-1990): COMPUTADORAS PERSONALES:
 - υ APARICION DE SOFTWARE **AMIGABLE CON EL USUARIO**: DESTINADO A USUARIOS NO PROFESIONALES Y CON UNA INTERFASE GRAFICA MUY DESARROLLADA.
 - υ DESARROLLO DE **SISTEMAS OPERATIVOS DE RED Y SISTEMAS OPERATIVOS DISTRIBUIDOS**.

HISTORIA DE LOS S. O. - GENERACIONES

- υ SISTEMAS OPERATIVOS DE RED:
 - Φ LOS USUARIOS ESTAN CONSCIENTES DE LA EXISTENCIA DE VARIAS COMPUTADORAS CONECTADAS.
 - Φ C/ MAQUINA EJECUTA SU PROPIO S. O. LOCAL.
 - Φ SON SIMILARES A LOS S. O. DE UN SOLO PROCESADOR PERO CON EL AGREGADO DE:
 - CONTROLADOR DE INTERFAZ DE LA RED Y SU SOFTWARE DE BAJO NIVEL.
 - SOFTWARE PARA CONEXION Y ACCESO A ARCHIVOS REMOTOS, ETC.
- υ SISTEMAS OPERATIVOS DISTRIBUIDOS:
 - Φ APARECE ANTE LOS USUARIOS COMO UN S. O. DE UN SOLO PROCESADOR, AUN CUANDO DE SOPORTE A VARIOS PROCESADORES.
 - Φ LOS USUARIOS NO SON CONSCIENTES DEL LUGAR DONDE SE EJECUTAN SUS PROGRAMAS O DONDE SE ENCUENTRAN SUS ARCHIVOS:
 - LO DEBE ADMINISTRAR EL S. O. AUTOMATICAMENTE.
 - Φ DEBEN PERMITIR QUE UN PROGRAMA SE EJECUTE MEDIANTE VARIOS PROCESADORES A LA VEZ, MAXIMIZANDO EL PARALELISMO.
- υ APARICION DE EMULADORES DE TERMINAL PARA EL ACCESO A EQUIPOS REMOTOS DESDE PC.
- υ GRAN ENFASIS EN LA **SEGURIDAD**, EN ESPECIAL POR EL DESARROLLO DE LOS SISTEMAS DE COMUNICACIONES DE DATOS.
- υ EL S. O. CREA UN AMBIENTE DE TRABAJO SEGUN EL CONCEPTO DE **MAQUINA VIRTUAL**, QUE LO AISLA DEL FUNCIONAMIENTO INTERNO DE LA MAQUINA.
- υ PROLIFERACION DE **SISTEMAS DE BASES DE DATOS**, ACCESIBLES MEDIANTE REDES DE COMUNICACION.

CONCEPTOS DE LOS SISTEMAS OPERATIVOS

- v LA INTERFAZ ENTRE EL S. O. Y LOS PROGRAMAS DEL USUARIO SE DEFINE COMO EL CONJUNTO DE “**INSTRUCCIONES AMPLIADAS**” QUE PROPORCIONA EL S. O.: SON LAS “**LLAMADAS AL SISTEMA**”:
 - u CREAN, ELIMINAN Y UTILIZAN **OBJETOS DEL SOFTWARE** CONTROLADOS POR EL S. O.:
 - Φ LOS MAS IMPORTANTES SON **PROCESOS Y ARCHIVOS**.
- v **PROCESOS:**
 - u ES EL CONCEPTO CENTRAL DE TODOS LOS S. O.
 - u ES BASICAMENTE UN PROGRAMA EN EJECUCION.
 - u CONSTA DEL PROGRAMA EJECUTABLE, SUS DATOS Y PILA, CONTADOR Y OTROS REGISTROS, ADEMAS DE LA INFORMACION NECESARIA PARA EJECUTAR EL PROGRAMA.
 - u LA INFORMACION DE CONTROL RELACIONADA CON LOS PROCESOS SE ALMACENA EN LA **TABLA DE PROCESOS**:
 - Φ ADMINISTRADA POR EL S. O.
 - Φ POSEE UN ARREGLO DE ESTRUCTURAS, UNA POR CADA PROCESO EXISTENTE EN ESE MOMENTO.
 - u UN PROCESO (SUSPENDIDO) CONSTA DE:
 - Φ UN ESPACIO DE DIRECCION.
 - Φ LOS DATOS PERTINENTES DE LA TABLA DE PROCESOS.
 - u UN PROCESO PUEDE CREAR **PROCESOS HIJO** Y ESTOS NUEVOS PROCESOS HIJO, CONFORMANDO UN **ARBOL DE PROCESOS**.
- v **ARCHIVOS:**
 - u UNA DE LAS FUNCIONES PRINCIPALES DEL S. O. ES BRINDAR INDEPENDENCIA DE DISPOSITIVO.
 - u MUCHOS S. O. SOPORTAN EL CONCEPTO DE **DIRECTORIO** COMO UNA FORMA DE AGRUPAR ARCHIVOS.
 - u LOS DIRECTORIOS SE ESTRUCTURAN JERARQUICAMENTE:
 - Φ A CADA ARCHIVO LE CORRESPONDE UNA **RUTA DE ACCESO**.
 - u EXISTEN DISTINTOS ESQUEMAS DE SEGURIDAD DE ARCHIVOS EN LOS DISTINTOS S. O.

CONCEPTOS DE LOS SISTEMAS OPERATIVOS

v LLAMADAS AL SISTEMA:

- u PERMITEN A LOS PROGRAMAS COMUNICARSE CON EL S. O. Y SOLICITARLE SERVICIOS.
- u A CADA LLAMADA LE CORRESPONDE UN PROCEDIMIENTO:
 - ⊕ PONE LOS PARAMETROS DE LA LLAMADA EN UN LUGAR ESPECIFICO PARA LUEGO EJECUTAR UNA INSTRUCCION TIPO “TRAP” DE LLAMADA A PROCEDIMIENTO PROTEGIDO PARA INICIAR EL S. O.
 - ⊕ LUEGO DE “TRAP” EL S. O. RECUPERA EL CONTROL , EXAMINA LOS PARAMETROS Y SI SON VALIDOS EJECUTA EL TRABAJO SOLICITADO.
 - ⊕ LUEGO DE TERMINAR, EL S. O. COLOCA UN CODIGO DE ESTADO EN UN REGISTRO INDICANDO SI TUVO EXITO O FRACASO Y EJECUTA UNA INSTRUCCION DEL TIPO “RETURN FROM TRAP” PARA REGRESAR EL CONTROL AL PROCEDIMIENTO.
 - ⊕ EL PROCEDIMIENTO REGRESA AL PROGRAMA LLAMADOR CON UN CODIGO DE ESTADO COMO UN VALOR DE FUNCION:
 - DENTRO DE LOS PARAMETROS PUEDEN REGRESAR VALORES ADICIONALES.

ESTRUCTURA DE LOS S. O.

v SE CONSIDERA LA **ORGANIZACION INTERNA** DE LOS S. O.

v **SISTEMAS MONOLITICOS:**

- u ES MUY COMUN: NO EXISTE ESTRUCTURA PROPIAMENTE DICHA O ES MINIMA.
- u EL S. O. ES UNA COLECCION DE PROCEDIMIENTOS QUE SE PUEDEN LLAMAR ENTRE SI.
- u CADA PROCEDIMIENTO TIENE UNA INTERFAZ BIEN DEFINIDA EN TERMINOS DE PARAMETROS Y RESULTADOS.
- u PARA EJECUTAR LOS SERVICIOS DEL S. O. (LLAMADAS AL SISTEMA):
 - ⊕ SE SOLICITAN COLOCANDO LOS PARAMETROS EN LUGARES BIEN DEFINIDOS (REGISTROS O PILAS).
 - ⊕ SE EJECUTA UNA INSTRUCCION ESPECIAL DE TRAMPA: **LLAMADA AL NUCLEO O LLAMADA AL SUPERVISOR.**
 - ⊕ LA INSTRUCCION CAMBIA LA MAQUINA DEL **MODO USUARIO AL MODO NUCLEO (O MODO SUPERVISOR).**
 - ⊕ SE TRANSFIERE EL CONTROL AL S. O.
 - ⊕ EL S. O. EXAMINA LOS PARAMETROS DE LA LLAMADA PARA DETERMINAR CUAL DE ELLAS SE DESEA REALIZAR.
 - ⊕ EL S. O. ANALIZA UNA TABLA QUE CONTIENE EN LA ENTRADA k UN APUNTADOR AL PROCEDIMIENTO QUE REALIZA LA k-ésima LLAMADA AL SISTEMA:
 - IDENTIFICA AL PROCEDIMIENTO DE SERVICIO LLAMADO.
 - ⊕ LA LLAMADA AL SISTEMA TERMINA Y EL CONTROL REGRESA AL PROGRAMA DEL USUARIO.

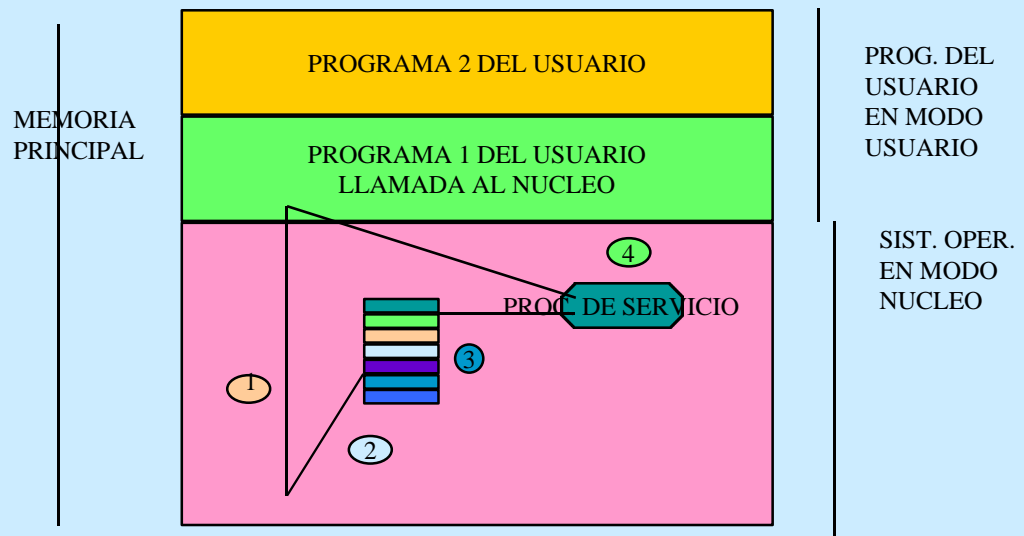
ESTRUCTURA DE LOS S. O.

v SISTEMAS CON CAPAS:

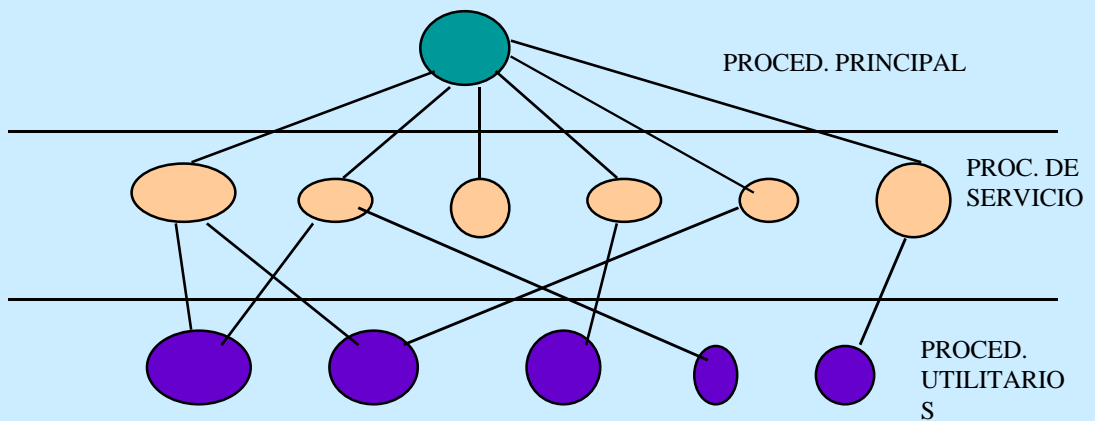
- u ES UNA GENERALIZACION DEL MODELO DE ESTRUCTURA SIMPLE PARA UN SISTEMA MONOLITICO.
- u CONSISTE EN ORGANIZAR EL S. O. COMO UNA JERARQUIA DE CAPAS, C/U CONSTRUIDA SOBRE LA INMEDIATA INFERIOR.

ESTRUCTURA DE LOS S. O.

FORMA EN QUE DEBE HACERSE UNA LLAMADA AL SISTEMA PARA UN SISTEMA MONOLITICO:

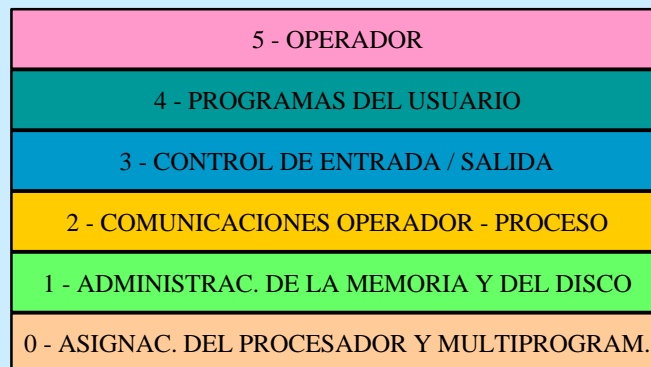


MODELO DE ESTRUCTURA SIMPLE PARA UN SISTEMA MONOLITICO:



ESTRUCTURA DE LOS S. O.

ESTRUCTURA DEL S. O. EN CAPAS “THE”:



- υ EL PRIMER SISTEMA CON ESTE ESQUEMA FUE EL “THE” (HOLANDA-DIJKSTRA-1968):
 - ⊕ “THE”: TECHNISCHE HOGESCHOOL EINDHOVEN.
 - ⊕ CAPA 0:
 - TRABAJA CON LA ASIGNACION DEL PROCESADOR.
 - ALTERNA ENTRE LOS PROCESOS CUANDO OCURREN LAS INTERRUPCIONES O EXPIRAN LOS CRONOMETROS.
 - PROPORCIONA LA MULTIPROGRAMACION BASICA.
 - ⊕ CAPA 1:
 - ADMINISTRA LA MEMORIA.
 - ASEGURA QUE LAS PAGINAS (PORCIONES DE MEMORIA) REQUERIDAS DE LOS PROCESOS LLEGUEN A MEMORIA CUANDO FUERAN NECESARIAS.
 - ⊕ CAPA 2:
 - ADMINISTRA LA COMUNICACION ENTRE CADA PROCESO Y LA CONSOLA DEL OPERADOR.
 - POR SOBRE ESTA CAPA, CADA PROCESO TIENE SU PROPIA CONSOLA DE OPERADOR.

ESTRUCTURA DE LOS S. O.

Φ CAPA 3:

- CONTROLA LOS DISPOSITIVOS DE E / S Y ALMACENA EN BUFFERS LOS FLUJOS DE INFORMACION ENTRE ELLOS.
- POR SOBRE LA CAPA 3 CADA PROCESO PUEDE TRABAJAR CON DISPOSITIVOS ABSTRACTOS DE E / S EN VEZ DE CON DISPOSITIVOS REALES.

Φ CAPA 4:

- ALOJA LOS PROGRAMAS DEL USUARIO.
- LOS PROG. DEL USUARIO NO TIENEN QUE PREOCUPARSE POR EL PROCESO, MEMORIA, CONSOLA O CONTROL DE E / S.

Φ CAPA 5:

- LOCALIZA EL PROCESO OPERADOR DEL SISTEMA.

υ UNA GENERALIZACION MAS AVANZADA DEL CONCEPTO DE CAPAS SE PRESENTO CON “MULTICS” (MIT, BELL LABS Y GENERAL ELECTRIC):

- Φ “MULTICS”: MULTIPLEXED INFORMATION AND COMPUTING SERVICE.
- Φ PRESENTA UNA ESTRUCTURA EN ANILLOS CONCENTRICOS, SIENDO LOS INTERIORES LOS PRIVILEGIADOS.
- Φ UN PROCEDIMIENTO DE UN ANILLO EXTERIOR, PARA LLAMAR A UN PROCEDIMIENTO DE UN ANILLO INTERIOR, DEBE HACER EL EQUIVALENTE A UNA LLAMADA AL SISTEMA.

υ MAQUINAS VIRTUALES:

- υ SE SEPARAN TOTALMENTE LAS FUNCIONES DE MULTIPROGRAMACION Y DE MAQUINA EXTENDIDA.
- υ EXISTE UN ELEMENTO CENTRAL LLAMADO **MONITOR DE LA MAQUINA VIRTUAL** QUE:
 - Φ SE EJECUTA EN EL HARDWARE.
 - Φ REALIZA LA MULTIPROGRAMACION.
 - Φ PROPORCIONA VARIAS MAQUINAS VIRTUALES A LA CAPA SUPERIOR.

ESTRUCTURA DE LOS S. O.

- υ LAS MAQUINAS VIRTUALES INSTRUMENTAN COPIAS “EXACTAS” DEL HARDWARE SIMPLE, CON SU MODO NUCLEO / USUARIO, E / S, INTERRUPCIONES Y TODO LO DEMAS QUE POSEE UNA MAQUINA REAL.
- υ PUEDEN EJECUTAR CUALQUIER S. O. QUE SE EJECUTE EN FORMA DIRECTA SOBRE EL HARDWARE.
- υ LAS DISTINTAS MAQUINAS VIRTUALES PUEDEN EJECUTAR DISTINTOS S. O. Y EN GENERAL ASI LO HACEN.
- υ SOPORTAN PERIFERICOS VIRTUALES.
- υ EJEMPLO DE S. O. REPRESENTATIVO DE ESTA ESTRUCTURA: “VM/370” DE IBM:
 - Φ LAS M. V. GENERALMENTE UTILIZARAN, ENTRE OTROS, EL S. O. “CMS”: CONVERSATIONAL MONITOR SYSTEM.
 - Φ CUANDO UN PROGRAMA “CMS” EJECUTA UNA LLAMADA AL SISTEMA:
 - LA LLAMADA ES ATRAPADA POR EL S. O. EN SU PROPIA M. V.; NO PASA DIRECTAMENTE AL “VM/370”.
 - “CMS” PROPORCIONA LAS INSTRUCCIONES DE E / S EN HARDWARE PARA LA LECTURA DEL DISCO VIRTUAL O LO NECESARIO PARA EFECTUAR LA LLAMADA.
 - “VM/370” ATRAPA ESTAS INSTRUCCIONES DE E / S Y LAS EJECUTA SOBRE EL HARDWARE VERDADERO.

υ MODELO CLIENTE-SERVIDOR:

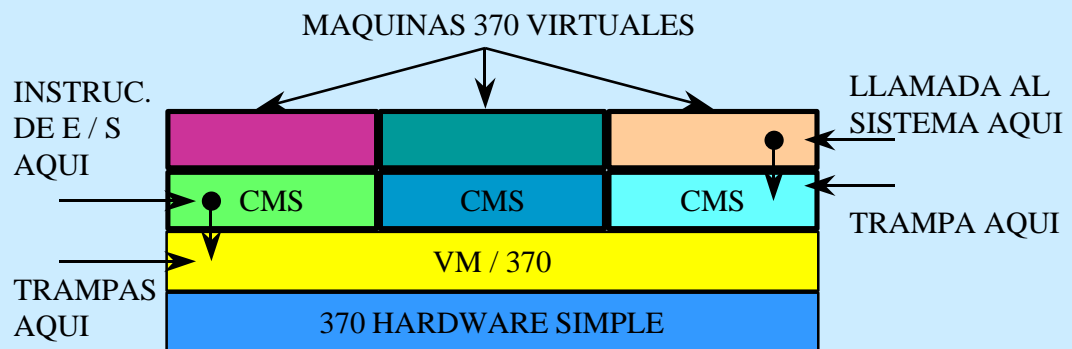
- υ UNA TENDENCIA EN LOS S. O. MODERNOS ES LA DE EXPLOTAR LA IDEA DE MOVER EL CODIGO A CAPAS SUPERIORES Y MANTENER UN **NUCLEO MINIMO**, DE MANERA SIMILAR AL “VM/370”.
- υ IMPLANTAR LA MAYORIA DE LAS FUNCIONES DEL S. O. EN LOS PROCESOS DEL USUARIO.
- υ PARA SOLICITAR UN SERVICIO (POR EJ.: LECTURA DE UN BLOQUE DE CIERTO ARCHIVO):
 - Φ EL PROCESO DEL USUARIO (**PROCESO CLIENTE**) ENVIA LA SOLICITUD A UN **PROCESO SERVIDOR**:
 - REALIZA EL TRABAJO Y REGRESA LA RESPUESTA.

ESTRUCTURA DE LOS S. O.

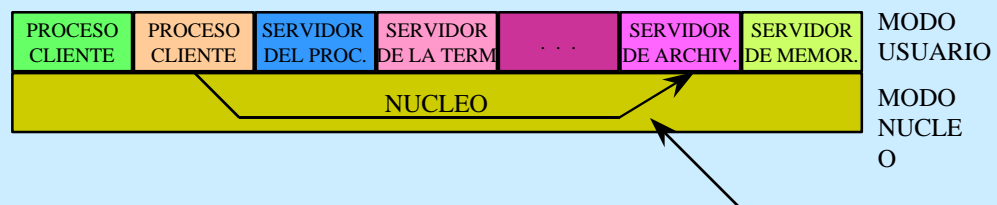
- υ EL NUCLEO CONTROLA LA COMUNICACION ENTRE LOS CLIENTES Y LOS SERVIDORES.
- υ SE FRACCIONA EL S. O. EN PARTES, C / U CONTROLANDO UNA FACETA:
 - Φ SERVICIO A ARCHIVOS, A PROCESOS, A TERMINALES, A MEMORIA, ETC., CADA PARTE PEQUEÑA Y MAS FACILMENTE CONTROLABLE.
- υ LOS SERVIDORES SE EJECUTAN COMO PROCESOS EN MODO USUARIO:
 - Φ NO TIENEN ACCESO DIRECTO AL HARDWARE.
 - Φ SE AISLAN Y ACOTAN MAS FACILMENTE LOS PROBLEMAS.
- υ SE ADAPTA PARA SU USO EN LOS SISTEMAS DISTRIBUIDOS:
 - Φ SI UN CLIENTE SE COMUNICA CON UN SERVIDOR MEDIANTE MENSAJES:
 - NO NECESITA SABER SI EL MENSAJE SE ATIENDE LOCALMENTE O MEDIANTE UN SERVIDOR REMOTO, SITUADO EN OTRA MAQUINA CONECTADA.
 - ENVIA UNA SOLICITUD Y OBTIENE UNA RESPUESTA.
- υ ALGUNAS FUNCIONES DEL S. O., POR EJ. EL CARGADO DE COMANDOS EN LOS REGISTROS FISICOS DEL DISPOSITIVO DE E / S, PRESENTAN PROBLEMAS ESPECIALES Y DISTINTAS SOLUCIONES:
 - Φ EJECUCION EN MODO NUCLEO, CON ACCESO TOTAL AL HARDWARE Y COMUNICACION CON LOS DEMAS PROCESOS MEDIANTE EL MECANISMO NORMAL DE MENSAJES.
 - Φ CONSTRUCCION DE UN MINIMO DE **MECANISMOS** DENTRO DEL NUCLEO MANTENIENDO LAS DECISIONES DE **POLITICA** RELATIVAS A LOS USUARIOS DENTRO DEL ESPACIO DEL USUARIO.

ESTRUCTURA DE LOS S. O.

LA ESTRUCTURA DE VM/370 CON CMS:

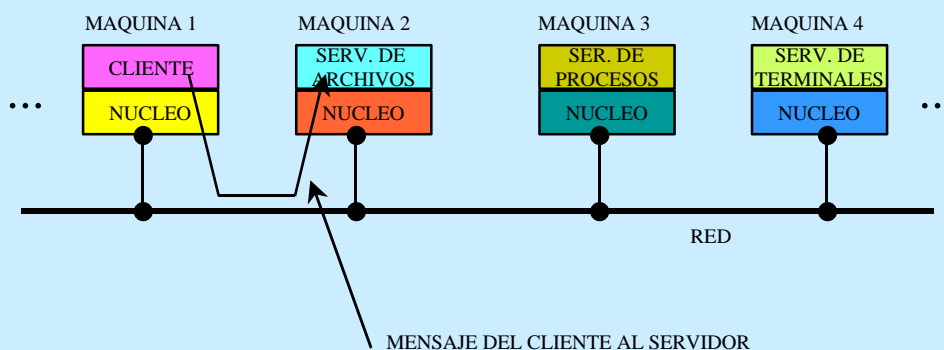


EL MODELO CLIENTE-SERVIDOR:



LOS CLIENTES OBTIENEN EL SERVICIO AL ENVIAR MENSAJES A LOS PROC. SERVIDORES

EL MODELO CLIENTE-SERVIDOR EN UN SISTEMA DISTRIBUIDO:



TENDENCIAS

- v SOPORTE GENERALIZADO PARA MULTIPROCESAMIENTO.
- v MIGRACION HACIA EL MICROCODIGO DE FUNCIONES DE LOS S. O. REALIZADAS POR SOFTWARE.
- v DISTRIBUCION DEL CONTROL ENTRE PROCESADORES LOCALIZADOS.
- v MEJORA DE LA EFICIENCIA EN EL SOPORTE DE LA EJECUCION CONCURRENTES DE PROGRAMAS.
- v SOPORTE DEL PARALELISMO MASIVO CON ALTISIMO GRADO DE CONCURRENCIA.
- v PROFUNDIZACION DE LOS ESQUEMAS DE MAQUINAS VIRTUALES.
- v CONTINUACION DEL ESQUEMA DE FAMILIAS DE S. O. PARA FAMILIAS DE COMPUTADORAS, VIENDO LAS APLICACIONES MAQUINAS VIRTUALES.
- v COMPATIBILIDAD CON NUEVAS GENERACIONES DE COMPUTADORAS.
- v DESARROLLOS EN LA INGENIERIA DE SOFTWARE PARA BRINDAR S. O. MAS PRESERVABLES, CONFIABLES Y COMPENSIBLES.
- v PROLIFERACION DE REDES DE SISTEMAS, DISTRIBUYENDO TAREAS EN EQUIPOS SOBRE LOS QUE EL USUARIO PUEDE NO TENER CONOCIMIENTO NI CONTROL:
 - u ENFASIS EN LA IMPORTANCIA DE LA PERSPECTIVA DE LAS MAQUINAS VIRTUALES.
- v PERMANENCIA DEL CONCEPTO DE ALMACENAMIENTO VIRTUAL.
- v PERMANENCIA DE LA PERSPECTIVA DEL S. O. COMO ADMINISTRADOR DE RECURSOS:
 - u LOS DATOS SERAN CONSIDERADOS CADA VEZ MAS COMO UN RECURSO PARA SER ADMINISTRADO.
- v PROFUNDIZACION DEL DESARROLLO DE S. O. CON FUNCIONES DISTRIBUIDAS ENTRE VARIOS PROCESADORES A TRAVES DE GRANDES REDES DE SISTEMAS.

HARDWARE

- v PRINCIPALES ASPECTOS DEL HARD. DE IMPORTANCIA PARA LOS S. O.:
- v **COMPAGINACION DEL ALMACENAMIENTO:**
 - u OBJETIVO: ACELERAR EL ACCESO AL ALMACENAMIENTO PRIMARIO (BANCOS DE MEMORIA).
 - u GENERALMENTE, MIENTRAS CUALQUIERA DE LAS LOCALIDADES DE UN **BANCO DE ALMACENAMIENTO PRIMARIO**, ESTA SIENDO ACCEDIDA, NINGUNA OTRA REFERENCIA PUEDE ESTAR EN CURSO.
 - u LA COMPAGINACION DEL ALMACENAMIENTO COLOCA LOCALIDADES DE MEMORIA ADYACENTES EN DIFERENTES BANCOS DE ALMACENAMIENTO, PARA PERMITIR VARIAS REFERENCIAS AL MISMO TIEMPO.
- v **REGISTRO DE RELOCALIZACION:**
 - u PERMITE RELOCALIZAR DE FORMA DINAMICA LOS PROGRAMAS.
 - u LA DIRECCION BASE DE UN PROGRAMA EN LA MEMORIA PRINCIPAL SE SITUA EN EL REGISTRO DE RELOCALIZACION.
 - u EL CONTENIDO DEL REG. DE RELOC. SE AÑADE A CADA DIRECCION DESARROLLADA POR UN PROGRAMA EN EJECUCION.
 - u PERMITE AL PROGRAMA RESIDIR EN LOCALIZACIONES DIFERENTES A AQUELLAS PARA LAS CUALES FUE TRADUCIDO.
- v **INTERRUPCIONES Y ESCRUTINIO:**
 - u INTERRUPCIONES: PERMITEN A UNA UNIDAD OBTENER LA INMEDIATA ATENCION DE OTRA, DE MANERA QUE LA PRIMERA PUEDA INFORMAR DE UN CAMBIO DE ESTADO:
 - Φ PERMITE SALVAR EL “ESTADO” DE LA UNIDAD INTERRUMPIDA ANTES DE PROCESAR LA INTERRUPCION.
 - u ESCRUTINIO: TECNICA QUE PERMITE QUE UNA UNIDAD VERIFIQUE EL ESTADO DE OTRA UNIDAD DE FUNCIONAMIENTO INDEPENDIENTE.
- v **UTILIZACION DEL “BUFFER”:**
 - u UN “BUFFER” ES UN AREA DE ALMACENAMIENTO PRIMARIO DESTINADA A CONTENER DATOS DURANTE TRANSFERENCIAS DE E / S.
 - u CUANDO CONCLUYE LA TRANSFERENCIA LOS DATOS PUEDEN SER ACCEDIDOS POR EL PROCESADOR.

HARDWARE

- υ ESQUEMA DE “ENTRADAS DE BUFFER SIMPLE”:
 - Φ EL CANAL DEPOSITA DATOS EN EL BUFFER.
 - Φ EL PROCESADOR PROCESA ESTOS DATOS.
 - Φ EL CANAL DEPOSITA NUEVOS DATOS, ETC.
 - Φ NO PUEDE HABER SIMULTANEIDAD ENTRE OPERACIONES DE COLOCAR DATOS EN EL BUFFER Y PROCESARLOS:
 - AFECTA LA PERFORMANCE.
- υ ESQUEMA DE “ENTRADAS DE BUFFER DOBLE”:
 - Φ PERMITE LA SOBREPOSICION DE OPERACIONES DE E / S CON EL PROCESAMIENTO:
 - MEJORA LA PERFORMANCE.
 - Φ MIENTRAS EL CANAL DEPOSITA DATOS EN UN BUFFER EL PROCESADOR PUEDE ESTAR PROCESANDO LOS DATOS DEL OTRO BUFFER.
 - Φ CUANDO EL PROCESADOR CONCLUYE EL PROCESO DE LOS DATOS DEL PRIMER BUFFER, PUEDE CONTINUAR CON LOS DATOS DEL SEGUNDO, MIENTRAS EL CANAL DEPOSITA NUEVOS DATOS EN EL PRIMER BUFFER:
 - Φ ES LA TECNICA DE “BUFFER BIESTABLE (O EN FLIP FLOP)”.
- v **DISPOSITIVOS PERIFERICOS:**
 - υ PERMITEN EL ALMACENAMIENTO DE GRANDES CANTIDADES DE INFORMACION FUERA DEL ALMACENAMIENTO PRINCIPAL.
 - υ EXISTEN DISPOSITIVOS SECUENCIALES Y DE ACCESO DIRECTO.
 - υ LAS CARACTERISTICAS Y PRESTACIONES SON MUY VARIADAS.
- v **PROTECCION DEL ALMACENAMIENTO:**
 - υ LIMITA EL N° DE DIRECCIONES QUE UN PROGRAMA PUEDE REFERENCIAR.
 - υ ES ESENCIAL EN LOS SISTEMAS MULTIUSUARIO.
 - υ SE IMPLEMENTA MEDIANTE LOS “REGISTROS DE LIMITES”: DEFINEN LAS DIRECCIONES SUPERIOR E INFERIOR DEL BLOQUE DE ALMACENAMIENTO AFECTADO A UN DETERMINADO PROGRAMA.

HARDWARE

- v TAMBIEN SE PUEDEN UTILIZAR “CLAVES DE PROTECCION DEL ALMACENAMIENTO” ANEXAS A AREAS DE ALMACENAMIENTO PRIMARIO:
 - Φ UN PROGRAMA SOLO PUEDE ACCEDER A LOCALIDADES DE ALMACENAMIENTO CUYAS CLAVES DE PROTECCION CONCUERDAN CON LAS DEL PROGRAMA.
- v **TEMPORIZADORES Y RELOJES:**
 - v “TEMPORIZADOR DE INTERVALOS”: PREVIENE QUE UN SOLO USUARIO MONOPOLICE EL PROCESADOR EN SIST. MULTIUSUARIO.
 - v EL TEMPORIZADOR GENERA UNA INTERRUPCION AL PROCESADOR CUANDO EXPIRA EL INTERVALO ASIGNADO A UN USUARIO.
 - v “RELOJ HORARIO”: PERMITE AL COMPUTADOR HACER UN SEGUIMIENTO DE LA “HORA DEL RELOJ DE PARED”, CON UNA EXACTITUD DE MILLONESIMAS DE SEGUNDO O MAYOR.
- v **OPERACIONES EN LINEA Y FUERA DE LINEA; PROCESADORES SATELITE:**
 - v “OPERACION EN LINEA”: LOS PERIFERICOS UTILIZADOS ESTAN CONECTADOS AL PROCESADOR.
 - v “OPERACION FUERA DE LINEA”: LOS PERIFERICOS UTILIZADOS ESTAN CONECTADOS A UNIDADES DE CONTROL QUE NO ESTAN CONECTADAS AL SISTEMA CENTRAL O PRINCIPAL.
- v **CANALES DE ENTRADA / SALIDA:**
 - v SON SISTEMAS COMPUTACIONALES DE PROPOSITO ESPECIAL, DEDICADOS AL MANEJO DE LA E / S CON INDEPENDENCIA DEL PROCESADOR PRINCIPAL.
 - v TIENEN ACCESO DIRECTO AL ALMACENAMIENTO PRINCIPAL PARA ALMACENAR O RECUPERAR INFORMACION.
 - v EVITAN AL PROCESADOR LA MAYOR PARTE DE LA CARGA DE MANEJAR LA E / S, INCREMENTANDO LA CONCURRENCIA.
 - v PRINCIPALES TIPOS:
 - Φ SELECTORES.
 - Φ MULTIPLEXORES DE BYTES.
 - Φ MULTIPLEXORES DE BLOQUES.

HARDWARE

v **ROBO DE CICLO:**

- u SIGNIFICA QUE EN LA COMPETENCIA ENTRE EL PROCESADOR Y LOS CANALES PARA ACCEDER A UN DETERMINADO BANCO DE ALMACENAMIENTO PRIMARIO (MEMORIA PRINCIPAL), SE DA PRIORIDAD A LOS CANALES:
 - Φ SE OPTIMIZA EL USO DE LOS DISPOSITIVOS DE E / S.

v **DIRECCIONAMIENTO DE BASE MAS DESPLAZAMIENTO:**

- u TODAS LAS DIRECCIONES SON AÑADIDAS AL CONTENIDO DE UN “REGISTRO DE BASE”.
- u LOS PROGRAMAS SON “INDEPENDIENTES DE LA LOCALIZACION”:
 - Φ ESPECIALMENTE IMPORTANTE EN AMBIENTES MULTIUSUARIO.

v **ESTADO DE PROBLEMA, ESTADO SUPERVISOR, INSTRUCCIONES PRIVILEGIADAS:**

- u CORRESPONDE A DISTINTOS “ESTADOS DE EJECUCION”.
- u “ESTADO DE PROBLEMA O DE USUARIO”: ESTADO EN QUE CORREN LOS PROGRAMAS DE USUARIO:
 - Φ TIENE ACCESO A UN SUBCONJUNTO DE INSTRUCCIONES DEL CONJUNTO DE INSTRUCCIONES DE LA MAQUINA.
- u “ESTADO SUPERVISOR O DE NUCLEO”: GENERALMENTE EL S. O. CORRE ASI CON LA CATEGORIA DE “USUARIO DE MAYOR CONFIANZA O NIVEL”:
 - Φ TIENE ACCESO A TODAS LAS INSTRUCCIONES DEL CONJUNTO DE INSTRUCCIONES DE LA MAQUINA.
- u SI EL SISTEMA SOPORTA MAS DE DOS ESTADOS:
 - Φ SE PUEDE INSTRUMENTAR UNA “GRANULACION DE PROTECCION” MAS FINA.
 - Φ PERMITE CONCEDER ACCESOS POR MEDIO DEL “PRINCIPIO DE MENOS PRIVILEGIO”:
 - SE DEBE GARANTIZAR A CADA USUARIO EN PARTICULAR LA MENOR CANTIDAD DE PRIVILEGIO Y ACCESO QUE NECESITE PARA CUMPLIR SUS TAREAS.
- u “INSTRUCCIONES PRIVILEGIADAS”: SON AQUELLAS A LAS QUE NO

HARDWARE

v ALMACENAMIENTO VIRTUAL:

- u LOS SISTEMAS DE ALMACENAMIENTO VIRTUAL PERMITEN A LOS PROGRAMAS REFERENCIAR DIRECCIONES QUE NO NECESITAN CORRESPONDER CON LAS DIRECCIONES REALES DISPONIBLES EN EL ALMACENAMIENTO PRIMARIO.
- u LAS “DIRECCIONES VIRTUALES” DESARROLLADAS POR LOS PROGRAMAS EN EJECUCION SON TRADUCIDAS DINAMICAMENTE POR EL HARDWARE A LAS “DIRECCIONES REALES” DE INSTRUCCIONES Y DATOS DEL ALMACENAMIENTO PRINCIPAL.
- u LOS PROGRAMAS PUEDEN REFERENCIAR ESPACIOS DE DIRECCIONES MUCHO MAYORES QUE LOS ESPACIOS DE DIRECCIONES DISPONIBLES EN EL ALMACENAMIENTO PRIMARIO.
- u SE UTILIZAN TECNICAS DE:
 - Φ “PAGINACION”: BLOQUES DE DATOS DE TAMAÑO FIJO VAN O VIENEN ENTRE EL ALMACENAMIENTO PRIMARIO Y EL SECUNDARIO.
 - Φ “SEGMENTACION”: IDENTIFICA LAS UNIDADES LOGICAS DE LOS PROGRAMAS Y DATOS PARA FACILITAR EL CONTROL DE ACCESO Y PARTICIPACION.

v MULTIPROCESAMIENTO:

- u VARIOS PROCESADORES COMPARTEN UN ALMACENAMIENTO PRIMARIO COMUN Y UN SOLO S. O.
- u ES NECESARIO “SECUENCIALIZAR” EL ACCESO A UNA LOCALIZACION (DIRECCION) DE ALMACENAMIENTO COMPARTIDO PARA QUE DOS O MAS PROCESADORES NO INTENTEN:
 - Φ MODIFICARLA AL MISMO TIEMPO.
 - Φ MODIFICARLA UNO(S) MIENTRAS OTRO(S) INTENTA(N) LEERLA.

v ACCESO DIRECTO A LA MEMORIA (DMA):

- u REQUIERE UNA SOLA INTERRUPCION AL PROCESADOR POR CADA BLOQUE DE CARACTERES TRANSFERIDOS DURANTE LA OPERACION DE E / S:
 - Φ MEJORA SIGNIFICATIVAMENTE LA PERFORMANCE (RENDIMIENTO).

HARDWARE

- v ES COMO SI EL PROCESADOR, EN VEZ DE INTERRUMPIDO FUERA RETRASADO.
- v MUY UTIL PARA ALTOS REQUERIMIENTOS DE E / S.
- v “CANAL DMA”: ES EL HARDWARE RESPONSABLE DEL ROBO DE CICLOS Y DE LA OPERACION DE LOS DISPOSITIVOS DE E / S.
- v **CANALIZACION:**
 - v TECNICA DE HARDWARE UTILIZADA PARA EXPLOTAR CIERTOS TIPOS DE PARALELISMO DURANTE EL PROCESAMIENTO DE INSTRUCCIONES.
 - v VARIAS INSTRUCCIONES PUEDEN ESTAR SIMULTANEAMENTE EN DIFERENTES ESTADOS DE EJECUCION.
- v **JERARQUIA DE ALMACENAMIENTO:**
 - v LOS NIVELES DE ALMACENAMIENTO INCLUYEN:
 - Φ ALMACENAMIENTO PRIMARIO: MEMORIA PRINCIPAL.
 - Φ ALMACENAMIENTO SECUNDARIO: DISCOS, CINTAS, ETC.
 - Φ ALMACENAMIENTO “CACHE”: MEMORIA MUY VELOZ DISEÑADA PARA AUMENTAR LA VELOCIDAD DE EJECUCION DE LOS PROGRAMAS:
 - ALOJA LA PARTE (INSTRUCCIONES Y DATOS) EN EJECUCION DE UN PROGRAMA.
 - v LOS NIVELES DE ALMACENAMIENTO CREAN “JERARQUIAS DE ALMACENAMIENTO”: CACHE, A. PRIMARIO, A. SECUNDARIO.
 - v AL BAJAR EN LA JERARQUIA:
 - Φ DESCIENDEN EL COSTO Y LA VELOCIDAD.
 - Φ AUMENTA LA CAPACIDAD.
 - v “ESPACIO DE DIRECCIONES”: CONJUNTO DE TODAS LAS DIRECCIONES DISPONIBLES PARA UN PROGRAMA.

SOFTWARE

- v CONSISTE EN LOS PROGRAMAS DE INSTRUCCIONES Y DATOS QUE DEFINEN PARA EL HARDWARE LOS ALGORITMOS NECESARIOS PARA LA RESOLUCION DE PROBLEMAS.
- v **PROGRAMACION EN LENGUAJE DE MAQUINA:**
 - u “LENGUAJE DE MAQUINA”:
 - ⊕ LENGUAJE DE PROGRAMACION QUE UN COMPUTADOR PUEDE COMPRENDER DIRECTAMENTE.
 - ⊕ ES “DEPENDIENTE DE LA MAQUINA”: UN PROGRAMA EN LENGUAJE DE MAQUINA ESCRITO EN EL COMPUTADOR DE UN FABRICANTE, GENERALMENTE NO PUEDE SER EJECUTADO EN EL DE OTRO, SALVO QUE SU LENGUAJE DE MAQUINA SEA COMPATIBLE.
 - ⊕ MUY POCO USADO ACTUALMENTE.
- v **ENSAMBLADORES Y MACROPROCESADORES:**
 - u LOS “LENGUAJES ENSAMBLADORES” SE DESARROLLARON PARA:
 - ⊕ INCREMENTAR LA VELOCIDAD DE PROGRAMACION .
 - ⊕ REDUCIR LOS ERRORES DE CODIFICACION.
 - u LOS PROGRAMAS DEBEN SER TRADUCIDOS AL “LENGUAJE DE MAQUINA” MEDIANTE UN PROGRAMA “ENSAMBLADOR”:
 - ⊕ TAMBIEN ES DEPENDIENTE DE LA MAQUINA.
 - u LOS “MACROPROCESADORES”:
 - ⊕ SE DESARROLLARON PARA ACELERAR LA CODIFICACION DE UN PROGRAMA ENSAMBLADOR.
 - ⊕ SE INCORPORARON EN LOS ENSAMBLADORES.
 - ⊕ UNA “MACROINSTRUCCION” INDICA LA EJECUCION DE VARIAS INSTRUCCIONES EN LENGUAJE ENSAMBLADOR.
 - ⊕ EL “PROCESADOR DE MACROINSTRUCCIONES” EFECTUA UNA “MACROEXPANSION” CUANDO LEE UNA MACRO DURANTE LA TRADUCCION DE UN PROGRAMA:
 - GENERA UNA SERIE DE INSTRUCCIONES EN LENGUAJE ENSAMBLADOR CORRESPONDIENTES A LA MACRO.

SOFTWARE

v **COMPILADORES:**

- u “LENGUAJES DE ALTO NIVEL”: SE DESARROLLARON PARA RESOLVER EL PROBLEMA DE LA DEPENDENCIA RESPECTO A LA MAQUINA.
- u PERMITEN EL DESARROLLO DE PROGRAMAS “INDEPENDIENTES DE LA MAQUINA”.
- u SE LOGRA: MAYOR VELOCIDAD DE PROGRAMACION, PROGRAMAS TRANSPORTABLES ENTRE SISTEMAS DIFERENTES Y MENORES REQUERIMIENTOS DE CONOCIMIENTOS DE HARDWARE.
- u “COMPILADORES”: TRADUCEN LOS LENGUAJES DE ALTO NIVEL AL LENGUAJE DE MAQUINA.
- u “TRADUCTORES”: DENOMINACION PARA “COMPILADORES” Y “ENSAMBLADORES”.
 - Φ ENTRADA: “PROGRAMA FUENTE” DEL PROGRAMADOR.
 - Φ SALIDA: “PROGRAMA OBJETO” O “PROGRAMA RESULTANTE”.

v **SISTEMAS DE CONTROL DE ENTRADA / SALIDA (IOCS: INPUT / OUTPUT CONTROL SYSTEM):**

- u EL IOCS LIBERA AL PROGRAMADOR DE APLICACIONES DE LA COMPLEJIDAD DE LA ADMINISTRACION DE LA E / S:
 - Φ PROGRAMAS DE CANAL, COORDINACION DE CANALES Y PROCESADORES, CONTROL DE LA E / S, ETC.
- u ES UNA MANIFESTACION DE LA TENDENCIA A QUE LOS DESARROLLADORES DE APLICACIONES SE CONCENTREN EN LA PRODUCCION DE CODIGOS ORIENTADOS HACIA LAS APLICACIONES Y NO HACIA LOS SISTEMAS (HARDWARE).

v **UTILIZACION DEL SPOOL (SIMULTANEOUS PERIPHERAL OPERATION ON LINE: OPERACION SIMULTANEA DE PERIFERICOS EN LINEA):**

- u UN DISPOSITIVO DE ALTA VELOCIDAD (EJ.: DISCO) SE INTERPONE ENTRE UN PROGRAMA EN EJECUCION Y UN DISPOSITIVO DE BAJA VELOCIDAD (EJ.: IMPRESORA) RELACIONADO CON EL PROGRAMA EN LA E / S.
- u EVITA LA DEMORA EN LA EJECUCION DE PROGRAMAS COMO CONSECUENCIA DEL USO DE PERIFERICOS LENTOS.

SOFTWARE

- v **LENGUAJES ORIENTADOS HACIA EL PROCEDIMIENTO *VERSUS* LENGUAJES ORIENTADOS HACIA EL PROBLEMA:**
 - u O. HACIA EL PROCEDIMIENTO: SON DE PROPOSITO GENERAL Y APTOS PARA RESOLVER GRAN VARIEDAD DE PROBLEMAS:
 - Φ EJ.: PASCAL, COBOL, FORTRAN, BASIC, PL/I.
 - u O. HACIA EL PROBLEMA: SON ESPECIFICOS PARA RESOLVER DETERMINADOS TIPOS DE PROBLEMAS:
 - Φ EJ.: GPSS (SIMULACION), SPSS (ESTADISTICA).
- v **COMPILADORES RAPIDOS Y SUCIOS *VERSUS* COMPILADORES OPTIMIZADORES:**
 - u C. RAPIDOS Y SUCIOS: PRODUCEN RAPIDAMENTE UN PROGRAMA OBJETO QUE PUEDE SER INEFICIENTE RESPECTO DE ALMACENAMIENTO Y VELOCIDAD DE EJECUCION:
 - Φ UTILES PARA EL DESARROLLO Y PRUEBA DE SISTEMAS.
 - u C. OPTIMIZADORES: PRODUCEN CON MAYOR LENTITUD UN CODIGO DE MAQUINA ALTAMENTE EFICIENTE EN ALMACENAMIENTO Y EJECUCION:
 - Φ UTILES EN ETAPA DE PRODUCCION DE LOS SISTEMAS.
- v **INTERPRETADORES:**
 - u NO PRODUCEN UN PROGRAMA OBJETO.
 - u EJECUTAN DIRECTAMENTE UN PROGRAMA FUENTE.
 - u SON UTILES EN AMBIENTES DE DESARROLLO DE PROGRAMAS.
 - u SON MAS LENTOS QUE LOS CODIGOS COMPILADOS.
- v **CARGADORES ABSOLUTOS Y DE RELOCALIZACION:**
 - u LOS PROGRAMAS SE EJECUTAN EN EL ALMACENAMIENTO PRINCIPAL.
 - u “ASIGNACION”: ES LA ASOCIACION DE INSTRUCCIONES Y DATOS CON LOCALIZACIONES PARTICULARES DE ALMACENAMIENTO.
 - u “CARGADOR”: ES UN PROGRAMA QUE COLOCA LAS INSTRUCCIONES Y DATOS DE UN PROGRAMA DENTRO DE LOCALIZACIONES DEL ALMACENAMIENTO PRINCIPAL.

SOFTWARE

- υ “CARGADOR ABSOLUTO”: COLOCA LAS INSTRUCCIONES Y DATOS EN LAS LOCALIZACIONES ESPECIFICAS INDICADAS EN EL PROGRAMA DE LENGUAJE DE MAQUINA.
- υ “CARGADOR DE RELOCALIZACION”: PUEDE CARGAR UN PROGRAMA EN VARIOS LUGARES DENTRO DEL ALMACENAMIENTO PRINCIPAL:
 - Φ DEPENDE DE LA DISPONIBILIDAD DE ALMACENAMIENTO PRIMARIO AL MOMENTO DE REALIZAR LA CARGA.
- υ “TIEMPO DE CARGA”: MOMENTO DE REALIZAR LA CARGA.
- v **CARGADORES DE ENLACE Y EDITORES DE ENLACE:**
 - υ EL PROGRAMA EN LENGUAJE DE MAQUINA PRODUCIDO POR UN TRADUCTOR DEBE SER COMBINADO CON OTROS PROGRAMAS EN LENGUAJE DE MAQUINA PARA FORMAR UNA UNIDAD EJECUTABLE.
 - υ LA “COMBINACION DE PROGRAMAS” ES REALIZADA POR “CARGADORES DE ENLACE” Y “EDITORES DE ENLACE” ANTES DEL TIEMPO DE EJECUCION DEL PROGRAMA.
 - υ “CARGADOR DE ENLACE”: EN EL MOMENTO DE CARGA, COMBINA CUALESQUIERA PROGRAMAS REQUERIDOS Y LOS CARGA DIRECTAMENTE EN EL ALMACENAMIENTO PRIMARIO.
 - υ “EDITOR DE ENLACE”: EJECUTA LA COMBINACION DE PROGRAMAS MENCIONADA Y ADEMAS CREA UNA IMAGEN DE CARGA A MEMORIA QUE PRESERVA EN EL ALMACENAMIENTO SECUNDARIO (DISCO), PARA USOS FUTUROS:
 - Φ ES MUY UTIL EN AMBIENTES DE PRODUCCION, YA QUE LA CARGA INMEDIATA DE LA IMAGEN DE MEMORIA PREVIAMENTE PRODUCIDA EVITA UN NUEVO PROCESO DE COMBINACION DE PROGRAMAS PREVIO A CADA EJECUCION.

MEMORIA FIJA

- v EL CONCEPTO DE “MICROPROGRAMACION” SUELE ATRIBUIRSE AL PROF. MAURICE WILKES (1951).
- v LA PRIMER APLICACION A GRAN ESCALA FUERON LOS S / 360 (IBM-'60).
- v “MICROPROGRAMACION DINAMICA”: PERMITE CARGAR FACILMENTE LOS NUEVOS “MICROPROGRAMAS” (“MICROCODIGO”) DENTRO DEL “ALMACENAMIENTO DE CONTROL”, DESDE DONDE SON EJECUTADOS:
 - u PERMITE VARIAR DINAMICA Y FRECUENTEMENTE LOS CONJUNTOS DE INSTRUCCIONES DE MAQUINA.
- v LA “MICROPROGRAMACION” INTRODUCE UNA CAPA DE PROGRAMACION POR DEBAJO DEL LENGUAJE DE MAQUINA:
 - u HACE POSIBLE DEFINIR LAS INSTRUCCIONES DEL LENGUAJE DE MAQUINA.
- v LOS “MICROPROGRAMAS” ESTAN FORMADOS POR “MICROINSTRUCCIONES” INDIVIDUALES QUE EN RELACION A LAS INSTRUCCIONES DE LOS LENGUAJES DE MAQUINA SON DE:
 - u NATURALEZA MUCHO MAS ELEMENTAL.
 - u FUNCION MAS DISPERSA.
- v CADA INSTRUCCION DE LENGUAJE DE MAQUINA ES IMPLEMENTADA POR UN MICROPROGRAMA COMPLETO QUE PUEDE SER EXTENSO:
 - u EL ALMACENAMIENTO DE CONTROL DEBE SER MUCHO MAS RAPIDO QUE EL ALMACENAMIENTO PRIMARIO.
- v **MICROCODIGOS VERTICAL Y HORIZONTAL:**
 - u M. VERTICAL:
 - Φ SIMILAR A LA EJECUCION DE INSTRUCCIONES EN LENGUAJE DE MAQUINA.
 - Φ ESPECIFICA EL MOVIMIENTO DE UNO O VARIOS DATOS ENTRE REGISTROS.
 - u M. HORIZONTAL:
 - Φ LAS MICROINSTRUCCIONES REQUIEREN MUCHOS MAS BITS.
 - Φ PUEDE ESPECIFICAR LA OPERACION PARALELA DE MOVIMIENTO DE DATOS ENTRE MUCHOS O TODOS LOS REGISTROS DE DATOS DE LA UNIDAD DE CONTROL.
 - Φ ES MAS PODEROSO PERO MAS COMPLEJO QUE EL M. VERTICAL.

MEMORIA FIJA

v **DECISION DE QUE FUNCIONES IMPLEMENTAR EN MICROCODIGO:**

- u UNA IMPORTANTE CUESTION DE DISEÑO ES DECIDIR QUE FUNCIONES DEL SISTEMA COMPUTACIONAL SE IMPLEMENTARAN EN MICROCODIGO.
- u EL MICROCODIGO PERMITE MEJORAR EL RENDIMIENTO EN LA EJECUCION DE UN SISTEMA COMPUTACIONAL.
- u EL CRITERIO FRECUENTEMENTE ES COLOCAR EN LA MEMORIA FIJA (EN VEZ DE EN EL SOFTWARE) LAS SECUENCIAS DE INSTRUCCIONES UTILIZADAS CON MAS FRECUENCIA.

v **EMULACION:**

- u ES UNA TECNICA POR MEDIO DE LA CUAL SE HACE QUE UNA MAQUINA APARENTE SER OTRA.
- u EL CONJUNTO DE INSTRUCCIONES DE LENGUAJE DE MAQUINA QUE VA A SER EMULADA SE MICROPROGRAMA EN LA “MAQUINA ANFITRIONA”.
- u LOS PROGRAMAS DE LENGUAJE DE MAQUINA DE LA MAQUINA EMULADA PUEDEN EJECUTARSE DIRECTAMENTE EN LA ANFITRIONA.
- u ES UTIL PARA COMPATIBILIDAD Y MIGRACION DE SISTEMAS.

v **MICRODIAGNOSTICOS:**

- u LOS MICROPROGRAMAS TIENEN MAS ACCESO AL HARDWARE QUE LOS PROGRAMAS DE LENGUAJE DE MAQUINA:
 - Φ ES POSIBLE EFECTUAR DETECCION Y CORRECCION DE ERRORES MAS AMPLIA A UN NIVEL MAS FINO.
- u SE PUEDE INTERCALAR EL “MICRODIAGNOSTICO” CON LAS INSTRUCCIONES DE PROGRAMAS DE LENGUAJE DE MAQUINA.

v **COMPUTADORES PERSONALIZADOS:**

- u EL HARDWARE PROPORCIONA UN AMBIENTE DE PROPOSITO GENERAL PARA EJECUTAR PROGRAMAS DE SOFTWARE:
 - Φ MOLDEAN EL SISTEMA COMPUTACIONAL SEGUN LAS NECESIDADES DEL USUARIO.
- u EN ALGUNOS SISTEMAS LOS USUARIOS PUEDEN EFECTUAR ESTA ADAPTACION POR MEDIO DEL MICROCODIGO.

MEMORIA FIJA

v ASISTENCIAS DE MICROCODIGO:

- u IMPLEMENTAN VARIAS RUTINAS DE MANEJO DE INTERRUPCIONES DE USO MAS FRECUENTE EN MICROCODIGO A FIN DE LOGRAR MEJORAS SIGNIFICATIVAS EN LA EJECUCION.

v MICROPROGRAMACION Y SISTEMAS OPERATIVOS:

- u FUNCIONES IMPLEMENTADAS FRECUENTEMENTE EN MICROCODIGO:
 - Φ MANEJO DE INTERRUPCIONES.
 - Φ MANTENIMIENTO DE VARIOS TIPOS DE ESTRUCTURAS DE DATOS.
 - Φ PRIMITIVAS DE SINCRONIZACION QUE CONTROLAN EL ACCESO A LOS DATOS COMPARTIDOS Y OTROS RECURSOS.
 - Φ OPERACIONES DE PALABRAS PARCIALES QUE PERMITEN QUE LAS OPERACIONES DE MANIPULACION DE BITS SEAN MANEJADAS EN FORMA EFICIENTE.
 - Φ “INTERCAMBIO DE CONTEXTO”, POR EJ., INTERCAMBIO RAPIDO DEL PROCESADOR ENTRE LOS USUARIOS DE UN SISTEMA DE USUARIOS MULTIPLES.
 - Φ SECUENCIAS DE REGRESO Y LLAMADA AL PROCEDIMIENTO.

PROCESOS Y ADMINISTRACION DEL PROCESADOR

- v INTRODUCCION Y DEFINICIONES SOBRE PROCESOS
- v ESTADOS DE PROCESO
- v PROCESAMIENTO DE INTERRUPCIONES
- v EL NUCLEO DEL S. O.
- v PLANIFICACION DE PROCESOS
- v NIVELES DE PLANIFICACION DEL PROCESADOR
- v OBJETIVOS DE LA PLANIFICACION
- v CRITERIOS DE PLANIFICACION
- v PLANIFICACION APROPIATIVA VERSUS NO APROPIATIVA
- v TEMPORIZADOR DE INTERVALOS O RELOJ DE INTERRUPCIONES
- v PRIORIDADES
- v TIPOS DE PLANIFICACION
- v MULTIPROCESAMIENTO
- v ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR
- v GRADOS DE ACOPLAMIENTO EN MULTIPROCESAMIENTO
- v S. O. DE MULTIPROCESADORES
- v RENDIMIENTO DEL SISTEMA DE MULTIPROCESAMIENTO
- v RECUPERACION DE ERRORES
- v MULTIPROCESAMIENTO SIMETRICO
- v TENDENCIAS DE LOS MULTIPROCESADORES

INTRODUCCION Y DEFINICIONES

SOBRE PROCESOS

- v EL CONCEPTO CENTRAL DE CUALQUIER S. O. ES EL DE “PROCESO”: UNA ABSTRACCION DE UN PROGRAMA EN EJECUCION TAMBIEN LLAMADA “TAREA”.
- v NO HAY UN ACUERDO UNIVERSAL SOBRE UNA DEFINICION DE PROCESO, PERO SI ALGUNAS DEFINICIONES ACEPTADAS:
 - v UN PROGRAMA QUE SE ESTA EJECUTANDO.
 - v UNA ACTIVIDAD ASINCRONICA.
 - v EL “EMPLAZAMIENTO DEL CONTROL” DE UN PROCEDIMIENTO QUE ESTA SIENDO EJECUTADO.
 - v AQUELLO QUE SE MANIFIESTA POR LA EXISTENCIA EN EL S. O. DE UN “BLOQUE DE CONTROL DE PROCESO”.
 - v AQUELLA ENTIDAD A LA CUAL SON ASIGNADOS LOS PROCESADORES.
 - v LA UNIDAD “DESPACHABLE”.
- v EN SISTEMAS DE **MULTIPROGRAMACION** LA CPU ALTERNA DE PROGRAMA EN PROGRAMA, EN UN ESQUEMA DE **SEUDOPARALELISMO**:
 - v LA CPU EJECUTA EN CIERTO INSTANTE UN SOLO PROGRAMA, INTERCAMBIANDO MUY RAPIDAMENTE ENTRE UNO Y OTRO.
- v EL **PARALELISMO REAL DE HARDWARE** SE DA:
 - v EN EJECUCION DE INSTRUCCIONES DE PROGRAMA CON MAS DE UN PROCESADOR DE INSTRUCCIONES EN USO SIMULTANEAMENTE.
 - v CON LA SUPERPOSICION DE EJECUCION DE INSTRUCCIONES DE PROGRAMA CON LA EJECUCION DE UNA O MAS OPERACIONES DE E / S.
- v EL OBJETIVO ES AUMENTAR EL PARALELISMO EN LA EJECUCION.
- v **EL MODELO DE PROCESOS**:
 - v TODO EL SOFTWARE EJECUTABLE, INCLUSIVE EL S. O., SE ORGANIZA EN VARIOS **PROCESOS SECUENCIALES O PROCESOS**.
 - v UN PROCESO INCLUYE AL PROGRAMA EN EJECUCION Y A LOS VALORES ACTIVOS DEL CONTADOR, REGISTROS Y VARIABLES DEL MISMO.
 - v CONCEPTUALMENTE CADA PROCESO TIENE SU PROPIA CPU VIRTUAL.

INTRODUCCION Y DEFINICIONES

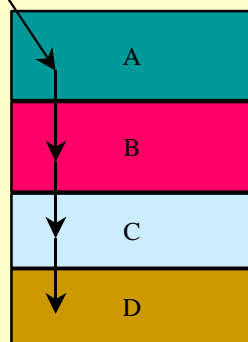
SOBRE PROCESOS

- υ SI LA CPU ALTERNA ENTRE LOS PROCESOS LA VELOCIDAD A LA QUE EJECUTA UN PROCESO NO SERA UNIFORME:
 - Φ LOS PROCESOS NO DEBEN PROGRAMARSE CON HIPOTESIS IMPLICITAS ACERCA DEL TIEMPO.
 - Φ NORMALMENTE LA MAYORIA DE LOS PROCESOS NO SON AFECTADOS POR LA MULTIPROGRAMACION SUBYACENTE DE LA CPU O LAS VELOCIDADES RELATIVAS DE PROCESOS DISTINTOS.
 - υ UN PROCESO ES UNA ACTIVIDAD DE UN CIERTO TIPO, QUE TIENE UN PROGRAMA, ENTRADA, SALIDA Y ESTADO.
 - υ UN SOLO PROCESADOR PUEDE SER COMPARTIDO ENTRE VARIOS PROCESOS CON CIERTO “ALGORITMO DE PLANIFICACION”:
 - Φ DETERMINA CUANDO DETENER EL TRABAJO EN UN PROCESO Y DAR SERVICIO A OTRO DISTINTO.
 - υ **JERARQUIAS DE PROCESOS:**
 - υ LOS S. O. DEBEN DISPONER DE UNA FORMA DE CREAR Y DESTRUIR PROCESOS CUANDO SE REQUIERA DURANTE LA OPERACION.
 - υ LOS PROCESOS PUEDEN GENERAR PROCESOS HIJOS MEDIANTE LLAMADAS AL S. O., PUDIENDO DARSE EJECUCION EN PARALELO.
 - υ **ESTADOS DEL PROCESO:**
 - υ CADA PROCESO ES UNA ENTIDAD INDEPENDIENTE PERO FRECUENTEMENTE DEBE INTERACTUAR CON OTROS PROCESOS.
 - υ LOS PROCESOS PUEDEN **BLOQUEARSE** EN SU EJECUCION PORQUE:
 - Φ DESDE EL PUNTO DE VISTA LOGICO NO PUEDE CONTINUAR (ESPERA DATOS QUE AUN NO ESTAN DISPONIBLES).
 - Φ EL S. O. ASIGNA LA CPU A OTRO PROCESO.
 - υ LOS ESTADOS QUE PUEDE TENER UN PROCESO SON:
 - Φ **EN EJECUCION:** UTILIZA LA CPU EN EL INSTANTE DADO.
 - Φ **LISTO:** EJECUTABLE, SE DETIENE EN FORMA TEMPORAL PARA QUE SE EJECUTE OTRO PROCESO.
 - Φ **BLOQUEADO:** NO SE PUEDE EJECUTAR DEBIDO A LA OCURRENCIA DE ALGUN EVENTO EXTERNO.
 - υ SON POSIBLES CUATRO **TRANSICIONES** ENTRE ESTOS ESTADOS.
- PROCESOS Y ADMINISTRACION DEL PROCESADOR

INTRODUCCION Y DEFINICIONES SOBRE PROCESOS

MULTIPROGRAMACION DE CUATRO PROGRAMAS

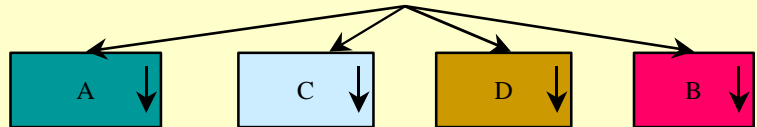
UN CONTADOR DE PROGRAMA



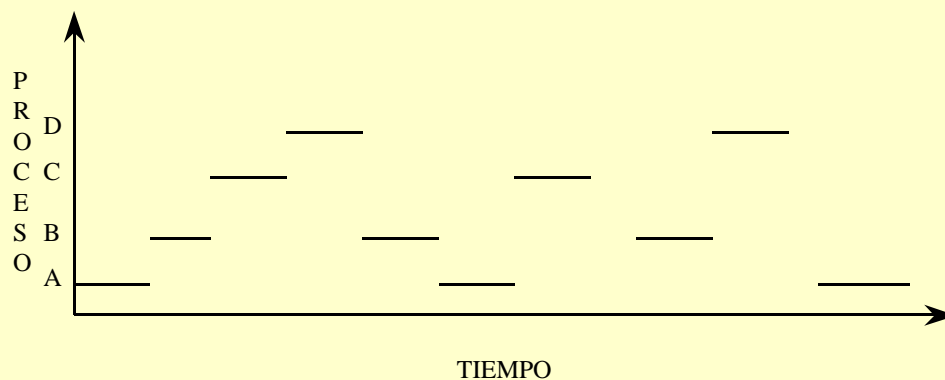
ALTERNADOR DE PROCESOS

MODELO CONCEPTUAL DE CUATRO PROCESOS SECUENCIALES INDEPENDIENTES

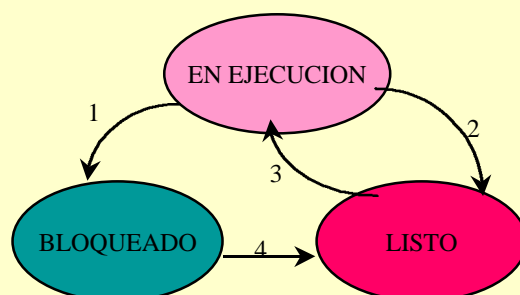
CUATRO CONTADORES DE PROGRAMA



SOLO UN PROGRAMA ESTA ACTIVO EN UN MOMENTO DADO



UN PROCESO PUEDE ESTAR EN EJECUCION, BLOQUEADO O LISTO



TANSICIONES ENTRE LOS ESTADOS

- 1-EL PROCESO SE BLOQUEA EN ESPERA DE DATOS
- 2-EL PLANIFICADOR ELIGE OTRO PROCESO
- 3-EL PLANIFICADOR ELIGE ESTE PROCESO
- 4-LOS DATOS ESTAN DISPONIBLES

- 1-BLOQUEO
- 2-TIEMPO EXCEDIDO
- 3-DESPACHO
- 4-DESPERTAR

ESTADOS DE PROCESOS

- v DURANTE SU EXISTENCIA UN PROCESO PASA POR UNA SERIE DE ESTADOS DISCRETOS:
 - u VARIAS CIRCUNSTANCIAS PUEDEN HACER QUE UN PROCESO CAMBIE DE ESTADO.
 - u SE PUEDE ESTABLECER UNA “LISTA DE LISTOS” PARA LOS PROCESOS LISTOS Y UNA “LISTA DE BLOQUEADOS” PARA LOS BLOQUEADOS.
 - u LA “LISTA DE LISTOS” SE MANTIENE EN ORDEN PRIORITARIO.
 - u LA “LISTA DE BLOQUEADOS” ESTA DESORDENADA:
 - Φ LOS PROCESOS SE DESBLOQUEAN EN EL ORDEN EN QUE TIENEN LUGAR LOS EVENTOS QUE ESTAN ESPERANDO.
- v AL ADMITIRSE UN TRABAJO EN EL SISTEMA SE CREA UN PROCESO EQUIVALENTE Y ES INSERTADO EN LA ULTIMA PARTE DE LA LISTA DE LISTOS.
- v LA ASIGNACION DE LA CPU AL PRIMER PROCESO DE LA “LISTA DE LISTOS” SE DENOMINA “DESPACHO”:
 - u ES EJECUTADO POR UNA ENTIDAD DEL S. O. LLAMADA “DESPACHADOR”.
- v EL BLOQUEO ES LA UNICA TRANSICION DE ESTADO INICIADA POR EL PROPIO PROCESO DEL USUARIO:
 - u LAS OTRAS TRANSICIONES SON INICIADAS POR ENTIDADES AJENAS AL PROCESO.
- v LA MANIFESTACION DE UN PROCESO EN UN S. O. ES UN “BLOQUE DE CONTROL DE PROCESO” (PCB) CON INFORMACION QUE INCLUYE:
 - u ESTADO ACTUAL DEL PROCESO.
 - u IDENTIFICACION UNICA DEL PROCESO.
 - u PRIORIDAD DEL PROCESO.
 - u APUNTADORES PARA LOCALIZAR LA MEMORIA DEL PROCESO.
 - u APUNTADORES PARA ASIGNAR RECURSOS.
 - u AREA PARA PRESERVAR REGISTROS.
- v CUANDO EL S. O. CAMBIA LA ATENCION DE LA CPU ENTRE LOS PROCESOS, UTILIZA LAS AREAS DE PRESERVACION DEL PCB PARA MANTENER LA INFORMACION QUE NECESITA PARA REINICIAR EL PROCESO CUANDO CONSIGA DE NUEVO LA CPU.

ESTADOS DE PROCESOS

- v LOS SISTEMAS QUE ADMINISTRAN LOS PROCESOS DEBEN PODER:
 - u CREAR, DESTRUIR, SUSPENDER, REANUDAR, CAMBIAR LA PRIORIDAD, BLOQUEAR, DESPERTAR Y DESPACHAR UN PROCESO.
- v LA CREACION DE UN PROCESO SIGNIFICA:
 - u DAR NOMBRE AL PROCESO.
 - u INSERTAR UN PROCESO EN LA LISTA DEL SISTEMA DE PROCESOS CONOCIDOS.
 - u DETERMINAR LA PRIORIDAD INICIAL DEL PROCESO.
 - u CREAR EL BLOQUE DE CONTROL DEL PROCESO.
 - u ASIGNAR LOS RECURSOS INICIALES DEL PROCESO.
- v UN PROCESO PUEDE “CREAR UN NUEVO PROCESO”:
 - u EL PROCESO CREADOR SE DENOMINA “PROCESO PADRE”.
 - u EL PROCESO CREADO SE DENOMINA “PROCESO HIJO”.
 - u SE OBTIENE UNA “ESTRUCTURA JERARQUICA DE PROCESOS”.
- v LA DESTRUCCION DE UN PROCESO IMPLICA:
 - u BORRARLO DEL SISTEMA.
 - u DEVOLVER SUS RECURSOS AL SISTEMA.
 - u PURGARLO DE TODAS LAS LISTAS O TABLAS DEL SISTEMA.
 - u BORRAR SU BLOQUE DE CONTROL DE PROCESOS.
- v UN PROCESO SUSPENDIDO NO PUEDE PROSEGUIR HASTA QUE OTRO PROCESO LO REANUDE.
- v REANUDAR (REACTIVAR) UN PROCESO IMPLICA REINICIARLO EN EL PUNTO DONDE FUE SUSPENDIDO.
- v LA DESTRUCCION DE UN PROCESO PUEDE O NO SIGNIFICAR LA DESTRUCCION DE LOS PROCESOS HIJOS, SEGUN EL S. O.
- v GENERALMENTE SE DENOMINA **TABLA DE PROCESOS** AL CONJUNTO DE INFORMACION DE CONTROL SOBRE LOS DISTINTOS PROCESOS.

PROCESAMIENTO DE INTERRUPCIONES

- v UNA “INTERRUPCION” ES UN EVENTO QUE ALTERA LA SECUENCIA EN QUE EL PROCESADOR EJECUTA LAS INSTRUCCIONES:
 - υ ES UN HECHO GENERADO POR EL HARDWARE DEL COMPUTADOR.
- v CUANDO OCURRE UNA INTERRUPCION EL S. O.:
 - υ OBTIENE EL CONTROL.
 - υ SALVA EL ESTADO DEL PROCESO INTERRUMPIDO:
 - Φ GENERALMENTE EN SU BLOQUE DE CONTROL DE PROCESOS.
 - υ ANALIZA LA INTERRUPCION.
 - υ TRANSFIERE EL CONTROL A LA RUTINA APROPIADA PARA LA MANIPULACION DE LA INTERRUPCION.
- v UNA INTERRUPCION PUEDE SER INICIADA POR:
 - υ UN PROCESO EN ESTADO DE EJECUCION.
 - υ UN EVENTO QUE PUEDE O NO ESTAR RELACIONADO CON UN PROCESO EN EJECUCION.
- v GENERALMENTE LAS **INTERRUPCIONES** SE PUEDEN CLASIFICAR POR **TIPOS** SEGUN EL SIGUIENTE DETALLE:
 - υ “SVC (LLAMADA AL SUPERVISOR)”:
 - Φ ES UNA PETICION GENERADA POR EL USUARIO PARA UN SERVICIO PARTICULAR DEL SISTEMA, POR EJ.:
 - REALIZACION DE E / S, OBTENCION DE MAS MEMORIA.
 - υ “E / S”:
 - Φ SON INICIADAS POR EL HARDWARE DE E / S, INDICANDO A LA CPU QUE HA CAMBIADO EL ESTADO DE UN CANAL O DISPOSITIVO, POR EJ.:
 - FINALIZACION DE E / S, OCURRENCIA DE UN ERROR.
 - υ “EXTERNAS”:
 - Φ SON CAUSADAS POR DISTINTOS EVENTOS, POR EJ.:
 - EXPIRACION DE UN CUANTO EN UN RELOJ DE INTERRUPCION.
 - RECEPCION DE UNA SEÑAL DE OTRO PROCESADOR EN UN SISTEMA MULTIPROCESADOR.

PROCESAMIENTO DE INTERRUPCIONES

- v “DE REINICIO”:
 - Φ OCURREN AL PRESIONAR LA “TECLA DE REINICIO” O CUANDO LLEGA UNA INSTRUCCION DE REINICIO DE OTRO PROCESADOR EN UN SISTEMA MULTIPROCESADOR.
- v “DE VERIFICACION DE PROGRAMA”:
 - Φ SON CAUSADAS POR ERRORES PRODUCIDOS DURANTE LA EJECUCION DE PROCESOS, POR EJ.:
 - UN INTENTO DE DIVIDIR POR CERO.
 - UN INTENTO DE UN PROCESO DE USUARIO DE EJECUTAR UNA INSTRUCCION PRIVILEGIADA.
 - UN INTENTO DE EJECUTAR UN CODIGO DE OPERACION INVALIDO.
- v “DE VERIFICACION DE MAQUINA”:
 - Φ SON OCASIONADAS POR UN MAL FUNCIONAMIENTO DEL HARDWARE.
- v EL S. O. INCLUYE RUTINAS LLAMADAS “MANIPULADORES DE INTERRUPCIONES (IH)” PARA PROCESAR CADA TIPO DIFERENTE DE INTERRUPCION.
- v CUANDO SE PRODUCE UNA INTERRUPCION EL S. O.:
 - v SALVA EL ESTADO DEL PROCESO INTERRUMPIDO.
 - v DIRIGE EL CONTROL AL MANIPULADOR DE INTERRUPCIONES ADECUADO.
 - v SE APLICA LA TECNICA DE “CAMBIO DE CONTEXTO”.
- v LOS S. O. INSTRUMENTAN INFORMACION DE CONTROL QUE PUEDE APARECER COMO LAS “PALABRAS DE ESTADO DE PROGRAMA (PSW)”:
 - v CONTROLAN EL ORDEN DE EJECUCION DE LAS INSTRUCCIONES.
 - v CONTIENEN INFORMACION SOBRE EL ESTADO DEL PROCESO.
- v EXISTEN TRES TIPOS DE PSW:
 - v ACTUAL, NUEVA Y VIEJA.

PROCESAMIENTO DE INTERRUPCIONES

- v LA “PSW ACTUAL”:
 - u ALMACENA LA DIRECCION DE LA PROXIMA INSTRUCCION QUE SERA EJECUTADA.
 - u INDICA LOS TIPOS DE INSTRUCCIONES ACTUALMENTE “HABILITADAS” E “INHABILITADAS”.
- v EN UN **SISTEMA UNIPROCESADOR** EXISTE:
 - u SOLO UNA PSW ACTUAL.
 - u SEIS PSW NUEVAS (UNA PARA CADA TIPO DE INTERRUPCION).
 - u SEIS PSW VIEJAS (UNA PARA CADA TIPO DE INTERRUPCION).
- v LA PSW NUEVA PARA UN TIPO DE INTERRUPCION DADO CONTIENE LA DIRECCION EN EL HARDWARE DONDE RESIDE EL MANIPULADOR DE INTERRUPCIONES PARA ESTE TIPO ESPECIFICO.
- v CUANDO OCURRE UNA INTERRUPCION PARA LA CUAL EL PROCESADOR NO ESTA INHABILITADO:
 - u EL HARDWARE CAMBIA LAS PSW EN LOS CASOS SIGUIENTES:
 - ⊕ AL ALMACENAR LA PSW ACTUAL EN LA PSW VIEJA, PARA ESTE TIPO DE INTERRUPCION.
 - ⊕ AL ALMACENAR LA PSW NUEVA EN LA PSW ACTUAL, PARA ESTE TIPO DE INTERRUPCION.
 - u LUEGO DE ESTE “INTERCAMBIO DE PSW”:
 - ⊕ LA PSW ACTUAL CONTIENE LA DIRECCION DEL MANIPULADOR DE INTERRUPCION ADECUADO.
 - ⊕ EL MANIPULADOR DE INTERRUPCIONES PROCESA LA INTERRUPCION.
 - ⊕ LUEGO DE PROCESAR LA INTERRUPCION LA CPU ES ENVIADA AL:
 - PROCESO QUE ESTABA EN EJECUCION EN EL MOMENTO DE LA INTERRUPCION.
 - PROCESO DE LISTO DE MAS ALTA PRIORIDAD.
 - ⊕ DEPENDE DE SI EL PROCESO DE INTERRUPCION ES:
 - “APROPIATIVO”: OBTIENE LA CPU SOLO SI NO HAY PROCESOS DE LISTOS.
 - “NO APROPIATIVO”: OBTIENE DE NUEVO LA CPU.

EL NUCLEO DEL SISTEMA OPERATIVO

- v EL “NUCLEO” DEL S. O. CONTROLA TODAS LAS OPERACIONES QUE IMPLICAN PROCESOS.
- v REPRESENTA SOLO UNA PEQUEÑA PORCION DEL CODIGO DE TODO EL S. O. PERO ES DE AMPLIO USO.
- v GENERALMENTE PERMANECE EN EL ALMACENAMIENTO PRIMARIO.
- v EL PROCESO DE INTERRUPCIONES SE INCLUYE EN EL NUCLEO:
 - u DEBE SER RAPIDO (ESPECIALMENTE EN SISTEMAS MULTIUSUARIO) PARA:
 - Φ OPTIMIZAR EL USO DE LOS RECURSOS DEL SISTEMA.
 - Φ PROVEER TIEMPOS DE RESPUESTA ACEPTABLES A LOS USUARIOS INTERACTIVOS.
- v EL NUCLEO INHABILITA LAS INTERRUPCIONES MIENTRAS RESPONDE A UNA INTERRUPCION:
 - u LAS INTERRUPCIONES SON HABILITADAS DE NUEVO DESPUES DE COMPLETAR EL PROCESO DE UNA INTERRUPCION.
- v EL NUCLEO DEL S. O. GENERALMENTE REALIZA LAS SIGUIENTES **FUNCIONES**:
 - u MANIPULACION DE INTERRUPCIONES.
 - u CREACION Y DESTRUCCION DE PROCESOS.
 - u CAMBIO DE ESTADOS DE PROCESOS.
 - u DESPACHO.
 - u SUSPENSION Y REANUDACION DE PROCESOS.
 - u SINCRONIZACION DE PROCESOS.
 - u COMUNICACION ENTRE PROCESOS.
 - u MANIPULACION DE BLOQUES DE CONTROL DE PROCESO.
 - u SOPORTE DE LAS ACTIVIDADES DE E / S.
 - u SOPORTE DE LA ASIGNACION Y DESASIGNACION DE ALMACENAMIENTO.
 - u SOPORTE DEL SISTEMA DE ARCHIVOS.
 - u SOPORTE DE UN MECANISMO DE LLAMADA / REGRESO AL PROCEDIMIENTO.
 - u SOPORTE DE CIERTAS FUNCIONES CONTABLES (ESTADISTICAS) DEL SISTEMA.

PLANIFICACION DE PROCESOS

- v CUANDO MAS DE UN PROCESO ES EJECUTABLE DESDE EL PUNTO DE VISTA LOGICO:
 - u EL S. O. DEBE DECIDIR CUAL DE ELLOS DEBE EJECUTARSE EN PRIMER TERMINO.
 - u EL **PLANIFICADOR** ES LA PORCION DEL S. O. QUE DECIDE.
 - u EL **ALGORITMO DE PLANIFICACION** ES EL UTILIZADO.
- v PRINCIPALES **CRITERIOS** RESPECTO DE UN BUEN ALGORITMO DE PLANIFICACION:
 - u “EQUIDAD”: GARANTIZAR QUE CADA PROCESO OBTIENE SU PROPORCION JUSTA DE LA CPU.
 - u “EFICACIA”: MANTENER OCUPADA LA CPU EL 100% DEL TIEMPO.
 - u “TIEMPO DE RESPUESTA”: MINIMIZAR EL TIEMPO DE RESPUESTA PARA LOS USUARIOS INTERACTIVOS.
 - u “TIEMPO DE REGRESO”: MINIMIZAR EL TIEMPO QUE DEBEN ESPERAR LOS USUARIOS POR LOTES (BATCH) PARA OBTENER SUS RESULTADOS.
 - u “RENDIMIENTO”: MAXIMIZAR EL N° DE TAREAS PROCESADAS POR HORA.
- v ALGUNAS DE ESTAS METAS SON CONTRADICTORIAS:
 - u EJ.: MINIMIZAR EL TIEMPO DE RESPUESTA PARA LOS USUARIOS INTERACTIVOS SIGNIFICARIA NO EJECUTAR LAS TAREAS BATCH.
- v CADA PROCESO ES UNICO E IMPREDECIBLE:
 - u PUEDEN REQUERIR INTENSIVAMENTE OPERACIONES DE E / S O INTENSIVAMENTE CPU.
 - u EL PLANIFICADOR DEL S. O. NO TIENE LA CERTEZA DE CUANTO TIEMPO TRANSCURRIRA HASTA QUE UN PROCESO SE BLOQUEE:
 - Φ POR UNA OPERACION DE E / S.
 - Φ POR OTRA RAZON.
- v PARA EVITAR QUE UN PROCESO SE APROPIE DE LA CPU UN TIEMPO EXCESIVO LOS EQUIPOS POSEEN UN DISPOSITIVO QUE PROVOCA UNA INTERRUPCION EN FORMA PERIODICA, POR EJ. 60 HZ:
 - u 60 VECES POR SEGUNDO.

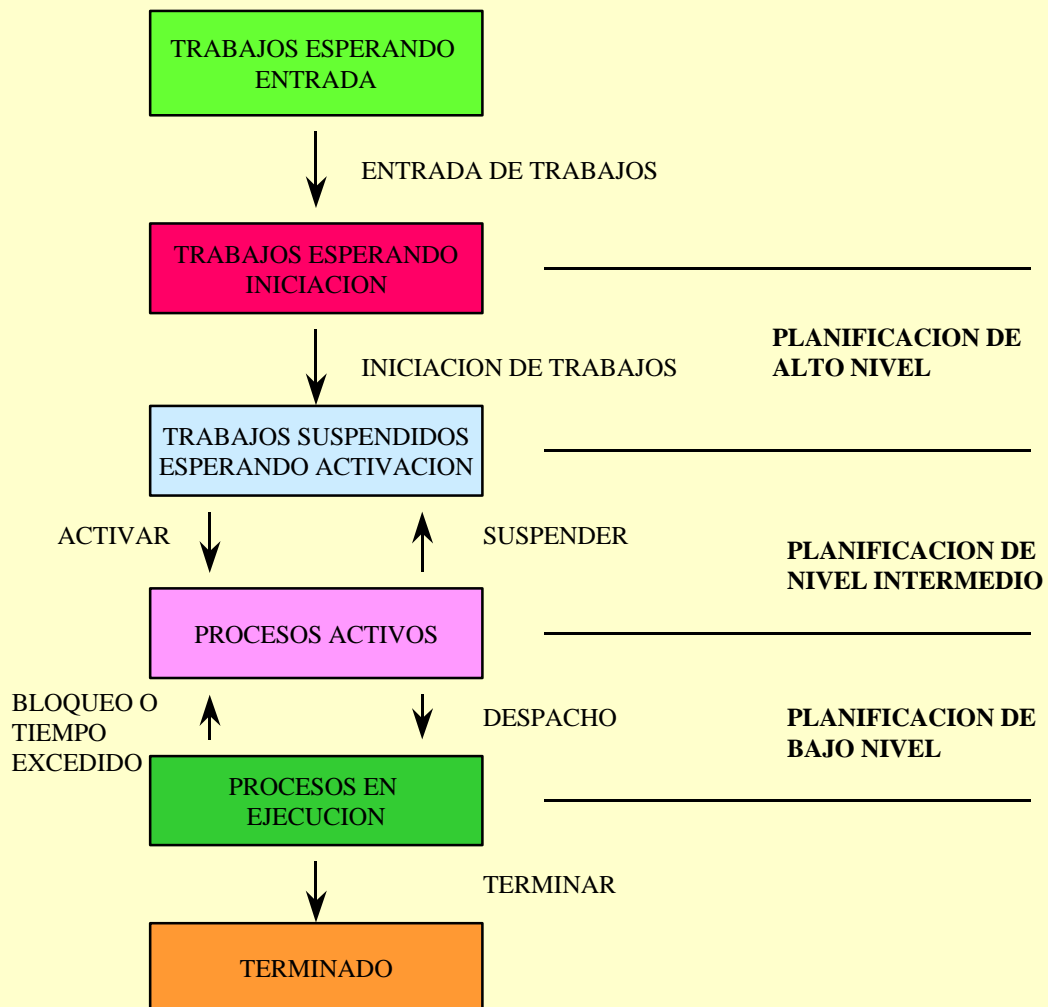
PLANIFICACION DE PROCESOS

- v EN CADA INTERRUPCION DEL RELOJ EL S. O. DECIDE:
 - u SI EL PROCESO QUE SE ESTA EJECUTANDO CONTINUA.
 - u SI EL PROCESO AGOTO SU TIEMPO DE CPU Y DEBE SUSPENDERSE Y CEDER LA CPU A OTRO PROCESO.
- v **PLANIFICACION APROPIATIVA:** ES LA ESTRATEGIA DE PERMITIR QUE PROCESOS EJECUTABLES (DESDE EL PUNTO DE VISTA LOGICO) SEAN SUSPENDIDOS TEMPORALMENTE.
- v **PLANIFICACION NO APROPIATIVA:** ES LA ESTRATEGIA DE PERMITIR LA EJECUCION DE UN PROCESO HASTA TERMINAR.
- v **PLANIFICACION DEL PROCESADOR:** DETERMINAR CUANDO DEBEN ASIGNARSE LOS PROCESADORES Y A QUE PROCESOS:
 - u ES RESPONSABILIDAD DEL S. O.

NIVELES DE PLANIFICACION DEL PROCESADOR

- v SE CONSIDERAN TRES NIVELES IMPORTANTES DE PLANIFICACION.
- v “PLANIFICACION DE ALTO NIVEL”:
 - u TAMBIEN SE DENOMINA “PLANIFICACION DE TRABAJOS”.
 - u DETERMINA A QUE TRABAJOS SE LES VA A PERMITIR COMPETIR ACTIVAMENTE POR LOS RECURSOS DEL SISTEMA:
 - ⊕ “PLANIFICACION DE ADMISION”.
- v “PLANIFICACION DE NIVEL INTERMEDIO”:
 - u DETERMINA A QUE PROCESOS SE LES PUEDE PERMITIR COMPETIR POR LA CPU.
 - u RESPONDE A FLUCTUACIONES A CORTO PLAZO EN LA CARGA DEL SISTEMA:
 - ⊕ EFECTUA “SUSPENSIONES” Y “ACTIVACIONES” (“REANUDACIONES”) DE PROCESOS.
 - u DEBE AYUDAR A ALCANZAR CIERTAS METAS EN EL RENDIMIENTO TOTAL DEL SISTEMA.
- v “PLANIFICACION DE BAJO NIVEL”:
 - u DETERMINA A QUE PROCESO LISTO SE LE ASIGNA LA CPU CUANDO ESTA QUEDA DISPONIBLE Y ASIGNA LA CPU AL MISMO:
 - ⊕ “DESPACHA” LA CPU AL PROCESO.
 - u LA EFECTUA EL “DESPACHADOR” DEL S. O.:
 - ⊕ OPERA MUCHAS VECES POR SEGUNDO.
 - ⊕ RESIDE SIEMPRE EN EL ALMACENAMIENTO PRIMARIO.
- v LOS DISTINTOS S. O. UTILIZAN VARIAS “POLITICAS DE PLANIFICACION”:
 - u SE INSTRUMENTAN MEDIANTE “MECANISMOS DE PLANIFICACION”.

NIVELES DE PLANIFICACION DEL PROCESADOR



OBJETIVOS DE LA PLANIFICACION

- v SER JUSTA:
 - u TODOS LOS PROCESOS SON TRATADOS DE IGUAL MANERA.
 - u NINGUN PROCESO ES POSTERGADO INDEFINIDAMENTE.
- v MAXIMIZAR LA CAPACIDAD DE EJECUCION:
 - u MAXIMIZAR EL N° DE PROCESOS SERVIDOS POR UNIDAD DE TIEMPO.
- v MAXIMIZAR EL N° DE USUARIOS INTERACTIVOS QUE RECIBAN UNOS TIEMPOS DE RESPUESTA ACEPTABLES:
 - u EN UN MAXIMO DE UNOS SEGUNDOS.
- v SER PREDECIBLE:
 - u UN TRABAJO DADO DEBE EJECUTARSE APROXIMADAMENTE EN LA MISMA CANTIDAD DE TIEMPO INDEPENDIENTEMENTE DE LA CARGA DEL SISTEMA.
- v MINIMIZAR LA SOBRECARGA:
 - u NO SUELE CONSIDERARSE UN OBJETIVO MUY IMPORTANTE.
- v EQUILIBRAR EL USO DE RECURSOS:
 - u FAVORECER A LOS PROCESOS QUE UTILIZARAN RECURSOS INFRAUTILIZADOS.
- v EQUILIBRAR RESPUESTA Y UTILIZACION:
 - u LA MEJOR MANERA DE GARANTIZAR BUENOS TIEMPOS DE RESPUESTA ES DISPONER DE RECURSOS SUFICIENTES CUANDO SE NECESITAN:
 - Φ LA UTILIZACION TOTAL DE RECURSOS PODRA SER POBRE.
- v EVITAR LA POSTERGACION INDEFINIDA:
 - u SE UTILIZA LA ESTRATEGIA DEL “ENVEJECIMIENTO”:
 - Φ MIENTRAS UN PROCESO ESPERA POR UN RECURSO SU PRIORIDAD DEBE AUMENTAR:
 - LA PRIORIDAD LLEGARA A SER TAN ALTA QUE EL PROCESO RECIBIRA EL RECURSO ESPERADO.
- v ASEGURAR LA PRIORIDAD:
 - u LOS MECANISMOS DE PLANIFICACION DEBEN FAVORECER A LOS PROCESOS CON PRIORIDADES MAS ALTAS.

OBJETIVOS DE LA PLANIFICACION

- v DAR PREFERENCIA A LOS PROCESOS QUE MANTIENEN RECURSOS CLAVES:
 - u UN PROCESO DE BAJA PRIORIDAD PODRIA MANTENER UN RECURSO CLAVE:
 - ⊕ PUEDE SER REQUERIDO POR UN PROCESO DE MAS ALTA PRIORIDAD.
 - ⊕ SI EL RECURSO ES NO APROPIATIVO:
 - EL MECANISMO DE PLANIFICACION DEBE OTORGAR AL PROCESO UN TRATAMIENTO MEJOR DEL QUE LE CORRESPONDERIA NORMALMENTE:
 - ES NECESARIO LIBERAR RAPIDAMENTE EL RECURSO CLAVE.
- v DAR MEJOR TRATAMIENTO A LOS PROCESOS QUE MUESTREN UN COMPORTAMIENTO DESEABLE:
 - u EJ.: TASA BAJA DE PAGINACION.
- v DEGRADARSE SUAVEMENTE CON CARGAS PESADAS:
 - u UN MECANISMO DE PLANIFICACION NO DEBE COLAPSAR CON EL PESO DE UNA EXIGENTE CARGA DEL SISTEMA.
 - u SE DEBE EVITAR UNA CARGA EXCESIVA:
 - ⊕ NO PERMITIENDO QUE SE CREEN NUEVOS PROCESOS CUANDO LA CARGA YA ES PESADA.
 - ⊕ DANDO SERVICIO A LA CARGA MAS PESADA AL PROPORCIONAR UN NIVEL MODERADAMENTE REDUCIDO DE SERVICIO A TODOS LOS PROCESOS.
- v MUCHAS DE ESTAS METAS SE ENCUENTRAN EN CONFLICTO ENTRE SI:
 - u LA PLANIFICACION SE CONVIERTE EN UN PROBLEMA COMPLEJO.

CRITERIOS DE PLANIFICACION

- v PARA REALIZAR LOS OBJETIVOS DE LA PLANIFICACION, UN MECANISMO DE PLANIFICACION DEBE CONSIDERAR LO SIGUIENTE:
 - v LA LIMITACION DE UN PROCESO A LAS OPERACIONES DE E / S:
 - u CUANDO UN PROCESO CONSIGUE LA CPU:
 - Φ ¿ LA UTILIZA SOLO BREVEMENTE ANTES DE GENERAR UNA PETICION DE E / S ?.
 - v LA LIMITACION DE UN PROCESO A LA CPU:
 - u CUANDO UN PROCESO OBTIENE LA CPU:
 - Φ ¿ TIENDE A USARLA HASTA QUE EXPIRA SU TIEMPO ?.
 - v SI UN PROCESO ES POR LOTE (BATCH) O INTERACTIVO:
 - u LOS USUARIOS INTERACTIVOS DEBEN RECIBIR INMEDIATO SERVICIO PARA GARANTIZAR BUENOS TIEMPOS DE RESPUESTA.
 - v ¿ QUE URGENCIA TIENE UNA RESPUESTA RAPIDA ?:
 - u EJ.: UN PROCESO DE TIEMPO REAL DE UN SISTEMA DE CONTROL QUE SUPERVISE UNA REFINERIA DE COMBUSTIBLE REQUIERE UNA RESPUESTA RAPIDA:
 - Φ MAS RAPIDA QUE LA RESPUESTA REQUERIDA POR UN PROCESO EN LOTES (BATCH) QUE DEBERA ENTREGARSE AL DIA SIGUIENTE.
 - v LA PRIORIDAD DE UN PROCESO:
 - u A MAYOR PRIORIDAD MEJOR TRATAMIENTO.
 - v FRECUENTEMENTE UN PROCESO GENERA FALLOS (CARENCIAS) DE PAGINA:
 - u PROBABLEMENTE LOS PROCESOS QUE GENERAN POCOS FALLOS DE PAGINA HAYAN ACUMULADO SUS CONJUNTOS DE TRABAJO EN EL ALMACENAMIENTO PRINCIPAL.
 - u LOS PROCESOS QUE EXPERIMENTAN GRAN CANTIDAD DE FALLOS DE PAGINA AUN NO HAN ESTABLECIDO SUS CONJUNTOS DE TRABAJO.
 - u UN CRITERIO INDICA FAVORECER A LOS PROCESOS QUE HAN ESTABLECIDO SUS CONJUNTOS DE TRABAJO.
 - u OTRO CRITERIO INDICA FAVORECER A LOS PROCESOS CON UNA TASA ALTA DE FALLOS DE PAGINA YA QUE RAPIDAMENTE GENERARAN UNA PETICION DE E / S.
- PROCESOS Y ADMINISTRACION DEL PROCESADOR 51

CRITERIOS DE PLANIFICACION

- v FRECUENTEMENTE UN PROCESO HA SIDO APROPIADO POR OTRO DE MAS ALTA PRIORIDAD:
 - u A MENUDO LOS PROCESOS APROPIADOS DEBEN RECIBIR UN TRATAMIENTO MENOS FAVORABLE.
 - u CADA VEZ QUE EL S. O. ASUME LA SOBRECARGA PARA HACER EJECUTAR ESTE PROCESO:
 - EL CORTO TIEMPO DE EJECUCION ANTES DE LA APROPIACION NO JUSTIFICA LA SOBRECARGA DE HACER EJECUTAR AL PROCESO EN PRIMER LUGAR.
- v ¿ CUANTO TIEMPO DE EJECUCION REAL HA RECIBIDO EL PROCESO ?:
 - u UN CRITERIO CONSIDERA QUE DEBE SER FAVORECIDO UN PROCESO QUE HA RECIBIDO MUY POCO TIEMPO DE CPU.
- v ¿ CUANTO TIEMPO ADICIONAL VA A NECESITAR EL PROCESO PARA TERMINAR ?:
 - u LOS TIEMPOS PROMEDIO DE ESPERA PUEDEN REDUCIRSE PRIORIZANDO LOS PROCESOS QUE REQUIEREN DE UN TIEMPO DE EJECUCION MINIMA PARA SU TERMINACION.
 - u POCAS VECES ES POSIBLE CONOCER LA CANTIDAD DE TIEMPO ADICIONAL QUE CADA PROCESO NECESITA PARA TERMINAR.

PLANIFICACION APROPIATIVA VERSUS NO APROPIATIVA

- v DISCIPLINA DE PLANIFICACION “APROPIATIVA”:
 - u UNA VEZ QUE SE LE HA OTORGADO LA CPU A UN PROCESO, LE PUEDE SER RETIRADA.
- v DISCIPLINA DE PLANIFICACION “NO APROPIATIVA”:
 - u UNA VEZ QUE SE LE HA OTORGADO LA CPU A UN PROCESO, NO LE PUEDE SER RETIRADA.
- v LA PLANIFICACION APROPIATIVA:
 - u ES UTIL CUANDO LOS PROCESOS DE ALTA PRIORIDAD REQUIEREN ATENCION RAPIDA.
 - u ES IMPORTANTE PARA GARANTIZAR BUENOS TIEMPOS DE RESPUESTA EN SISTEMAS INTERACTIVOS DE TIEMPO COMPARTIDO.
 - u TIENE SU COSTO EN RECURSOS:
 - Φ EL INTERCAMBIO DE CONTEXTO IMPLICA SOBRECARGA.
 - Φ REQUIERE MANTENER MUCHOS PROCESOS EN EL ALMACENAMIENTO PRINCIPAL, EN ESPERA DE LA CPU:
 - IMPLICA SOBRECARGA.
- v LA PLANIFICACION NO APROPIATIVA:
 - u SIGNIFICA QUE LOS TRABAJOS “LARGOS” HACEN ESPERAR A LOS TRABAJOS “CORTOS”.
 - u LOGRA MAS EQUIDAD EN EL TRATAMIENTO DE LOS PROCESOS.
 - u LOGRA HACER MAS PREDECIBLES LOS TIEMPOS DE RESPUESTA:
 - Φ LOS TRABAJOS NUEVOS DE PRIORIDAD ALTA NO PUEDEN DESPLAZAR A LOS TRABAJOS EN ESPERA.
- v EL DISEÑO DE UN MECANISMO APROPIATIVO HACE NECESARIO CONSIDERAR LAS ARBITRARIEDADES DE CASI CUALQUIER ESQUEMA DE PRIORIDADES:
 - u MUCHAS VECES LAS PROPIAS PRIORIDADES NO SON ASIGNADAS DE FORMA SIGNIFICATIVA.
- v EL MECANISMO DEBERIA SER SENCILLO PERO EFECTIVO Y SIGNIFICATIVO.

TEMPORIZADOR DE INTERVALOS O RELOJ DE INTERRUPCION

- v EL PROCESO AL CUAL ESTA ASIGNADA LA CPU SE DICE QUE ESTA EN EJECUCION:
 - υ PUEDE SER UN PROCESO DE S. O. O DE USUARIO.
- v EL S. O. DISPONE DE MECANISMOS PARA QUITARLE LA CPU A UN PROCESO DE USUARIO PARA EVITAR QUE MONOPOLICE EL SISTEMA.
- v EL S. O. POSEE UN “RELOJ DE INTERRUPCION” O “TEMPORIZADOR DE INTERVALOS” PARA GENERAR UNA INTERRUPCION:
 - υ EN ALGUN TIEMPO FUTURO ESPECIFICO O
 - υ DESPUES DE UN TRANCURSO DE TIEMPO EN EL FUTURO.
 - υ LA CPU ES ENTONCES DESPACHADA HACIA EL SIGUIENTE PROCESO.
- v UN PROCESO RETIENE EL CONTROL DE LA CPU HASTA QUE:
 - υ LA LIBERA VOLUNTARIAMENTE.
 - υ EL RELOJ LA INTERRUMPE.
 - υ ALGUNA OTRA INTERRUPCION ATRAE LA ATENCION DE LA CPU.
- v SI EL RELOJ INTERRUMPE UN PROCESO DE USUARIO:
 - υ LA INTERRUPCION CAUSA LA EJECUCION DEL S. O.
 - υ EL S. O. DECIDE CUAL SERA EL PROCESO QUE OBTENDRA LA CPU.
- v EL RELOJ DE INTERRUPCION AYUDA A GARANTIZAR TIEMPOS DE RESPUESTA RAZONABLES A USUARIOS INTERACTIVOS:
 - υ EVITA QUE EL SISTEMA SE “CUELQUE” A UN SOLO USUARIO EN UN CICLO INFINITO.
 - υ PERMITE QUE LOS PROCESOS RESPONDAN A “EVENTOS DEPENDIENTES DEL TIEMPO”.
- v LOS PROCESOS QUE NECESITAN UNA EJECUCION PERIODICA DEPENDEN DEL RELOJ DE INTERRUPCION.

PRIORIDADES

- v LAS PRIORIDADES PUEDEN SER:
 - u ASIGNADAS AUTOMATICAMENTE POR EL SISTEMA.
 - u ASIGNADAS DESDE EL EXTERIOR.
 - u DINAMICAS.
 - u ESTATICAS.
 - u ASIGNADAS RACIONALMENTE.
 - u ASIGNADAS ARBITRARIAMENTE:
 - Φ UN MECANISMO DEL SISTEMA NECESITA DISTINGUIR ENTRE PROCESOS SIN IMPORTARLE CUAL ES EL MAS IMPORTANTE.
- v **PRIORIDADES ESTATICAS VERSUS DINAMICAS:**
 - u LAS “PRIORIDADES ESTATICAS”:
 - Φ NO CAMBIAN.
 - Φ LOS MECANISMOS DE IMPLEMENTACION SON SENCILLOS.
 - Φ IMPLICAN UNA SOBRECARGA RELATIVAMENTE BAJA.
 - Φ NO RESPONDEN A CAMBIOS EN EL AMBIENTE (CONTEXTO) QUE HARIAN DESEABLE AJUSTAR ALGUNA PRIORIDAD.
 - u LAS “PRIORIDADES DINAMICAS”:
 - Φ RESPONDEN AL CAMBIO.
 - Φ LA PRIORIDAD INICIAL ASIGNADA A UN PROCESO PUEDE DURAR POCO TIEMPO:
 - LUEGO SE LA REAJUSTA A UN MEJOR VALOR.
 - Φ LOS MECANISMOS DE IMPLEMENTACION SON MAS COMPLICADOS QUE PARA PRIORIDADES ESTATICAS.
 - Φ IMPLICAN UNA SOBRECARGA MAYOR QUE PARA ESQUEMAS ESTATICOS.
- v **PRIORIDADES ADQUIRIDAS:**
 - u HACE REFERENCIA AL TRATAMIENTO ESPECIAL QUE EN SITUACIONES EXCEPCIONALES REQUIERE UN CIERTO PROCESO:
 - Φ PUEDE SIGNIFICAR RESTAR RECURSOS A LOS RESTANTES PROCESOS.

TIPOS DE PLANIFICACION

- v **PLANIFICACION A PLAZO FIJO:**
- v CIERTOS TRABAJOS SE PLANIFICAN PARA SER TERMINADOS EN UN TIEMPO ESPECIFICO O PLAZO FIJO.
- v ES UNA PLANIFICACION COMPLEJA DEBIDO A:
 - u EL USUARIO DEBE SUMINISTRAR ANTICIPADAMENTE UNA LISTA PRECISA DE RECURSOS NECESARIOS PARA EL PROCESO:
 - Φ GENERALMENTE NO SE DISPONE DE DICHA INFORMACION.
 - u LA EJECUCION DEL TRABAJO DE PLAZO FIJO NO DEBE PRODUCIR UNA GRAVE DEGRADACION DEL SERVIO A OTROS USUARIOS.
 - u EL SISTEMA DEBE PLANIFICAR CUIDADOSAMENTE SUS NECESIDADES DE RECURSOS HASTA EL PLAZO FIJO:
 - Φ SE PUEDE COMPLICAR CON LAS DEMANDAS DE RECURSOS DE NUEVOS PROCESOS QUE INGRESEN AL SISTEMA.
 - u LA CONCURRENCIA DE VARIOS PROCESOS DE PLAZO FIJO (ACTIVOS A LA VEZ) PUEDE REQUERIR METODOS SOFISTICADOS DE OPTIMIZACION.
 - u LA ADMINISTRACION INTENSIVA DE RECURSOS PUEDE GENERAR UNA CONSIDERABLE SOBRECARGA ADICIONAL.
- v **PLANIFICACION GARANTIZADA:**
- v SE ESTABLECEN COMPROMISOS DE DESEMPEÑO CON EL PROCESO DEL USUARIO:
 - u EJ.: SI EXISTEN “N” PROCESOS EN EL SISTEMA EL PROCESO DEL USUARIO RECIBIRA CERCA DEL “1/N” DE LA POTENCIA DE LA CPU.
- v EL SISTEMA DEBE TENER UN REGISTRO DEL:
 - u TIEMPO DE CPU QUE CADA PROCESO HA TENIDO DESDE SU ENTRADA AL SISTEMA.
 - u TIEMPO TRANSCURRIDO DESDE ESA ENTRADA.
- v CON LOS DATOS ANTERIORES Y EL REGISTRO DE PROCESOS EN CURSO DE EJECUCION EL SISTEMA:
 - u CALCULA Y DETERMINA QUE PROCESOS ESTAN MAS ALEJADOS POR DEFECTO DE LA RELACION “1/N” PROMETIDA.
 - u PRIORIZA LOS PROCESOS QUE HAN RECIBIDO MENOS CPU DE LA PROMETIDA.

TIPOS DE PLANIFICACION

- v **PLANIFICACION DEL PRIMERO EN ENTRAR PRIMERO EN SALIR (FIFO).**
- v ES MUY SIMPLE:
 - LOS PROCESOS SE DESPACHAN DE ACUERDO CON SU TIEMPO DE LLEGADA A LA COLA DE LISTOS.
- v UNA VEZ QUE EL PROCESO OBTIENE LA CPU SE EJECUTA HASTA TERMINAR:
 - ES UNA DISCIPLINA “NO APROPIATIVA”.
- v PUEDE OCASIONAR QUE:
 - PROCESOS LARGOS HAGAN ESPERAR A PROCESOS CORTOS.
 - PROCESOS NO IMPORTANTES HAGAN ESPERAR A PROCESOS IMPORTANTES.
- v ES MAS PREDECIBLE QUE OTROS ESQUEMAS.
- v NO PUEDE GARANTIZAR BUENOS TIEMPOS DE RESPUESTA INTERACTIVOS.
- v SUELE UTILIZARSE INTEGRADO A OTROS ESQUEMAS:
 - LOS PROCESOS SE DESPACHAN CON ALGUN ESQUEMA DE PRIORIDAD.
 - LOS PROCESOS CON IGUAL PRIORIDAD SE DESPACHAN “FIFO”.
- v **PLANIFICACION DE ASIGNACION EN RUEDA (RR: ROUND ROBIN):**
- v LOS PROCESOS:
 - SE DESPACHAN EN “FIFO”.
 - DISPONEN DE UNA CANTIDAD LIMITADA DE TIEMPO DE CPU:
 - Φ “DIVISION DE TIEMPO” O “CUANTO”.
- v SI UN PROCESO NO TERMINA ANTES DE EXPIRAR SU TIEMPO DE CPU:
 - LA CPU ES APROPIADA.
 - LA CPU ES OTORGADA AL SIGUIENTE PROCESO EN ESPERA.
 - EL PROCESO APROPIADO ES SITUADO AL FINAL DE LA LISTA DE LISTOS.
- v ES EFECTIVA EN AMBIENTES DE TIEMPO COMPARTIDO.
- v LA SOBRECARGA DE LA APROPIACION SE MANTIENE BAJA MEDIANTE MECANISMOS EFICIENTES DE INTERCAMBIO DE CONTEXTO Y CON PROCESOS Y ADMINISTRACION DEL PROCESADOR

TIPOS DE PLANIFICACION

- v **TAMAÑO DEL CUANTO O QUANTUM:**
 - v LA DETERMINACION DEL TAMAÑO DEL CUANTO ES DECISIVA PARA LA OPERACION EFECTIVA DE UN SISTEMA COMPUTACIONAL.
 - v LOS INTERROGANTES SON:
 - v ¿ CUANTO PEQUEÑO O GRANDE ?.
 - v ¿ CUANTO FIJO O VARIABLE ?.
 - v ¿ CUANTO IGUAL PARA TODOS LOS PROCESOS DE USUARIOS O DETERMINADO POR SEPARADO PARA C / U DE ELLOS ?.
 - v SI EL CUANTO SE HACE MUY GRANDE:
 - v CADA PROCESO RECIBE TODO EL TIEMPO NECESARIO PARA LLEGAR A SU TERMINACION:
 - Φ LA ASIGNACION EN RUEDA (“RR”) DEGENEREA EN “FIFO”.
 - v SI EL CUANTO SE HACE MUY PEQUEÑO:
 - v LA SOBRECARGA DEL INTERCAMBIO DE CONTEXTO SE CONVIERTE EN UN FACTOR DOMINANTE.
 - v EL RENDIMIENTO DEL SISTEMA SE DEGRADA:
 - Φ LA MAYOR PARTE DEL TIEMPO DE CPU SE INVIERTE EN EL INTERCAMBIO DEL PROCESADOR (CAMBIO DE CONTEXTO).
 - Φ LOS PROCESOS DE USUARIO DISPONEN DE MUY POCO TIEMPO DE CPU.
 - v EL CUANTO DEBE SER LO SUFICIENTEMENTE GRANDE COMO PARA PERMITIR QUE LA GRAN MAYORIA DE LAS PETICIONES INTERACTIVAS REQUIERAN DE MENOS TIEMPO QUE LA DURACION DEL CUANTO:
 - v EL TIEMPO TRANSCURRIDO DESDE EL OTORGAMIENTO DE LA CPU A UN PROCESO HASTA QUE GENERA UNA PETICION DE E / S DEBE SER MENOR QUE EL CUANTO ESTABLECIDO:
 - Φ OCURRIDA LA PETICION LA CPU PASA A OTRO PROCESO.
 - Φ COMO EL CUANTO ES MAYOR QUE EL TIEMPO TRANSCURRIDO HASTA LA PETICION DE E / S:
 - LOS PROCESOS TRABAJAN AL MAXIMO DE VELOCIDAD.
 - SE MINIMIZA LA SOBRECARGA DE APROPIACION.
 - SE MAXIMIZA LA UTILIZACION DE LA E / S.
- PROCESOS Y ADMINISTRACION DEL PROCESADOR

TIPOS DE PLANIFICACION

- v EL CUANTO OPTIMO VARIA DE UN SISTEMA A OTRO Y CON LA CARGA.
- v UN VALOR DE REFERENCIA ES 100 MSEG (MILISEGUNDOS).
- v **PLANIFICACION DEL TRABAJO MAS CORTO PRIMERO (SJF):**
- v ES UNA DISCIPLINA NO APROPIATIVA Y POR LO TANTO NO RECOMENDABLE EN AMBIENTES DE TIEMPO COMPARTIDO.
- v EL PROCESO EN ESPERA CON EL MENOR TIEMPO ESTIMADO DE EJECUCION HASTA SU TERMINACION ES EL SIGUIENTE EN EJECUTARSE.
- v LOS TIEMPOS PROMEDIO DE ESPERA SON MENORES QUE CON “FIFO”.
- v LOS TIEMPOS DE ESPERA SON MENOS PREDECIBLES QUE EN “FIFO”.
- v FAVORECE A LOS PROCESOS CORTOS EN DETRIMENTO DE LOS LARGOS.
- v TIENDE A REDUCIR EL N° DE PROCESOS EN ESPERA Y EL N° DE PROCESOS QUE ESPERAN DETRAS DE PROCESOS LARGOS.
- v REQUIERE UN CONOCIMIENTO PRECISO DEL TIEMPO DE EJECUCION DE UN PROCESO, LO QUE GENERALMENTE SE DESCONOCE.
- v SE PUEDEN ESTIMAR LOS TIEMPOS EN BASE A SERIES DE VALORES ANTERIORES.
- v **PLANIFICACION DEL TIEMPO RESTANTE MAS CORTO (SRT):**
- v ES LA CONTRAPARTE APROPIATIVA DEL SJF.
- v ES UTIL EN SISTEMAS DE TIEMPO COMPARTIDO.
- v EL PROCESO CON EL TIEMPO ESTIMADO DE EJECUCION MENOR PARA FINALIZAR ES EL SIGUIENTE EN SER EJECUTADO.
- v UN PROCESO EN EJECUCION PUEDE SER APROPIADO POR UN NUEVO PROCESO CON UN TIEMPO ESTIMADO DE EJECUCION MENOR.
- v TIENE MAYOR SOBRECARGA QUE LA PLANIFICACION SJF.
- v DEBE MANTENER UN REGISTRO DEL TIEMPO DE SERVICIO TRANSCURRIDO DEL PROCESO EN EJECUCION:
 - v AUMENTA LA SOBRECARGA.
- v LOS TRABAJOS LARGOS TIENEN UN PROMEDIO Y UNA VARIANZA DE LOS TIEMPOS DE ESPERA AUN MAYOR QUE EN SJF.
- v LA APROPIACION DE UN PROCESO A PUNTO DE TERMINAR POR OTRO DE MENOR DURACION RECIEN LLEGADO PODRIA SIGNIFICAR UN MAYOR TIEMPO DE CAMBIO DE CONTEXTO (ADMINISTRACION DEL PROCESADOR) QUE EL TIEMPO DE FINALIZACION DEL PRIMERO.

TIPOS DE PLANIFICACION

- v **AL DISEÑARSE LOS S. O. SE DEBE CONSIDERAR CUIDADOSAMENTE LA SOBRECARGA DE LOS MECANISMOS DE ADMINISTRACION DE RECURSOS COMPARANDOLA CON LOS BENEFICIOS ESPERADOS.**
- v **PLANIFICACION EL SIGUIENTE CON RELACION DE RESPUESTA MAXIMA (HRN):**
- v **CORRIGE ALGUNAS DE LAS DEBILIDADES DEL SJF:**
 - u EXCESO DE PERJUICIO HACIA LOS PROCESOS (TRABAJOS) LARGOS.
 - u EXCESO DE FAVORITISMO HACIA LOS NUEVOS TRABAJOS CORTOS.
- v **ES UNA DISCIPLINA NO APROPIATIVA.**
- v **LA PRIORIDAD DE CADA PROCESO ESTA EN FUNCION:**
 - u NO SOLO DEL TIEMPO DE SERVICIO DEL TRABAJO.
 - u TAMBIEN INFLUYE LA CANTIDAD DE TIEMPO QUE EL TRABAJO HA ESTADO ESPERANDO SER SERVIDO.
- v **CUANDO UN PROCESO HA OBTENIDO LA CPU CORRE HASTA TERMINAR.**
- v **LAS PRIORIDADES, QUE SON DINAMICAS, SE CALCULAN SEGUN:**
 - u $PRIORIDAD = (TE + TS) / TS$.
 - u TE: TIEMPO DE ESPERA; TS: TIEMPO DE SERVICIO.
- v **PLANIFICACION POR PRIORIDAD:**
- v **CONSIDERA FACTORES EXTERNOS AL PROCESO.**
- v **LAS IDEAS CENTRALES SON:**
 - u CADA PROCESO TIENE ASOCIADA UNA PRIORIDAD.
 - u EL PROCESO EJECUTABLE CON MAXIMA PRIORIDAD ES EL QUE TIENE EL PERMISO DE EJECUCION.
- v **LOS PROCESOS DE ALTA PRIORIDAD PODRIAN EJECUTAR INDEFINIDAMENTE:**
 - u EL PLANIFICADOR DEL SISTEMA PUEDE DISMINUIR LA PRIORIDAD DEL PROCESO EN EJECUCION EN CADA INTERRUPCION DEL RELOJ.
- v **LAS PRIORIDADES TAMBIEN PUEDEN SER ASIGNADAS DINAMICAMENTE POR EL SISTEMA PARA LOGRAR CIERTAS METAS:**
 - u RELACIONADAS CON EL PROCESADOR O LA E / S.

TIPOS DE PLANIFICACION

- v LOS PROCESOS LIMITADOS POR LA E / S (REQUERIMIENTOS INTENSIVOS DE E / S) OCUPAN MUCHO DE SU TIEMPO EN ESPERA DE E / S:
 - u DEBEN TENER PRIORIDAD PARA USAR LA CPU Y EFECTUAR LA SIGUIENTE PETICION DE E / S:
 - Φ SE EJECUTARA (LA E / S) EN PARALELO CON OTRO PROCESO QUE UTILICE LA CPU.
 - u SI DEBEN ESPERAR MUCHO TIEMPO A LA CPU ESTARAN OCUPANDO MEMORIA POR UN TIEMPO INNECESARIO.
- v UN ALGORITMO SENCILLO CONSISTE EN ESTABLECER QUE LA PRIORIDAD SEA 1 / F:
 - u “F” ES LA FRACCION DEL ULTIMO CUANTO UTILIZADO POR EL PROCESO.
 - u UN PROCESO QUE UTILICE 2 MSEG DE SU CUANTO DE 100 MSEG TENDRA PRIORIDAD 50.
 - u UN PROCESO QUE SE EJECUTO 50 MSEG ANTES DEL BLOQUEO TENDRA PRIORIDAD 2.
 - u UN PROCESO QUE UTILIZO TODO EL CUANTO TENDRA PRIORIDAD 1.
- v FRECUENTEMENTE LOS PROCESOS SE AGRUPAN EN “CLASES DE PRIORIDAD”:
 - u SE UTILIZA LA PLANIFICACION CON PRIORIDADES ENTRE LAS CLASES Y CON ROUND ROBIN (RR) DENTRO DE CADA CLASE.
 - u SI LAS PRIORIDADES NO SE REAJUSTAN EN ALGUN MOMENTO:
 - Φ LOS PROCESOS DE LAS CLASES DE PRIORIDAD MINIMA PODRIAN DEMORARSE INDEFINIDAMENTE.
- v **COLAS DE RETROALIMENTACION DE NIVELES MULTIPLES:**
- v PROPORCIONAN UNA ESTRUCTURA PARA LOGRAR:
 - u FAVORECER TRABAJOS CORTOS.
 - u FAVORECER TRABAJOS LIMITADOS POR LA E / S PARA OPTIMIZAR EL USO DE LOS DISPOSITIVOS DE E / S.
 - u DETERMINAR LA NATURALEZA DE UN TRABAJO LO MAS RAPIDO POSIBLE Y PLANIFICAR EL TRABAJO (PROCESO) EN CONSECUENCIA.

TIPOS DE PLANIFICACION

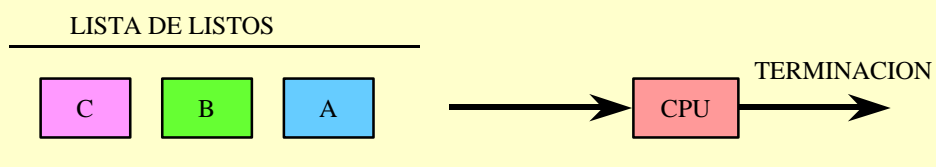
- v UN NUEVO PROCESO ENTRA EN LA RED DE LINEA DE ESPERA AL FINAL DE LA COLA SUPERIOR.
- v SE MUEVE POR ESTA COLA “FIFO” HASTA OBTENER LA CPU.
- v SI EL TRABAJO TERMINA O ABANDONA LA CPU PARA ESPERAR POR LA TERMINACION DE UNA OPERACION DE E / S O LA TERMINACION DE ALGUN OTRO SUCESO:
 - v EL TRABAJO ABANDONA LA RED DE LINEA DE ESPERA.
- v SI SU CUANTO EXPIRA ANTES DE ABANDONAR LA CPU VOLUNTARIAMENTE:
 - v EL PROCESO SE COLOCA EN LA PARTE TRASERA DE LA COLA DEL SIGUIENTE NIVEL INFERIOR.
- v EL TRABAJO RECIBE SERVICIO AL LLEGAR A LA CABEZA DE ESTA COLA SI LA PRIMERA ESTA VACIA.
- v MIENTRAS EL PROCESO CONTINUE CONSUMIENDO TOTALMENTE SU CUANTO EN CADA NIVEL:
 - v CONTINUARA MOVIENDOSE HACIA EL FINAL DE LAS COLAS INFERIORES.
- v GENERALMENTE HAY UNA COLA EN LA PARTE MAS PROFUNDA A TRAVES DE LA CUAL EL PROCESO CIRCULA EN ASIGNACION DE RUEDA HASTA QUE TERMINA.
- v EXISTEN ESQUEMAS EN LOS QUE EL CUANTO OTORGADO AL PROCESO AUMENTA A MEDIDA QUE EL PROCESO SE MUEVE HACIA LAS COLAS DE LOS NIVELES INFERIORES:
 - v CUANTO MAS TIEMPO HAYA ESTADO EL PROCESO EN LA RED DE LINEA DE ESPERA:
 - ❖ MAYOR SERA SU CUANTO CADA VEZ QUE OBTIENE LA CPU.
 - ❖ NO PODRA OBTENER LA CPU MUY A MENUDO DEBIDO A LA MAYOR PRIORIDAD DE LOS PROCESOS DE LAS COLAS SUPERIORES.
- v UN PROCESO SITUADO EN UNA COLA DADA NO PODRA SER EJECUTADO HASTA QUE LAS COLAS DE LOS NIVELES SUPERIORES ESTEN VACIAS.
- v UN PROCESO EN EJECUCION ES APROPIADO POR UN PROCESO QUE LLEGUE A UNA COLA SUPERIOR.

PROCESOS Y ADMINISTRACION DEL PROCESADOR

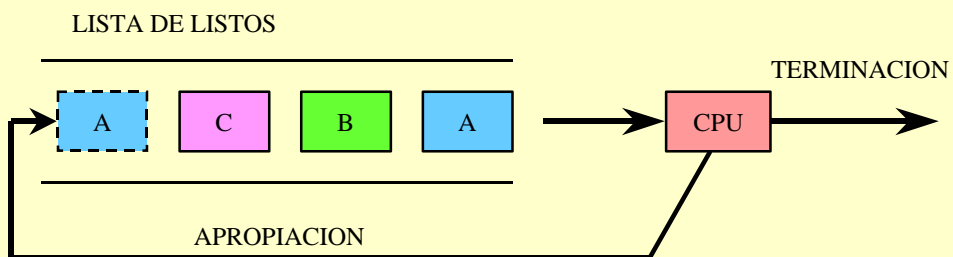
- v ES UN “MECANISMO ADAPTABLE”: SE ADAPTA A CARGAS VARIABLES. 62

TIPOS DE PLANIFICACION

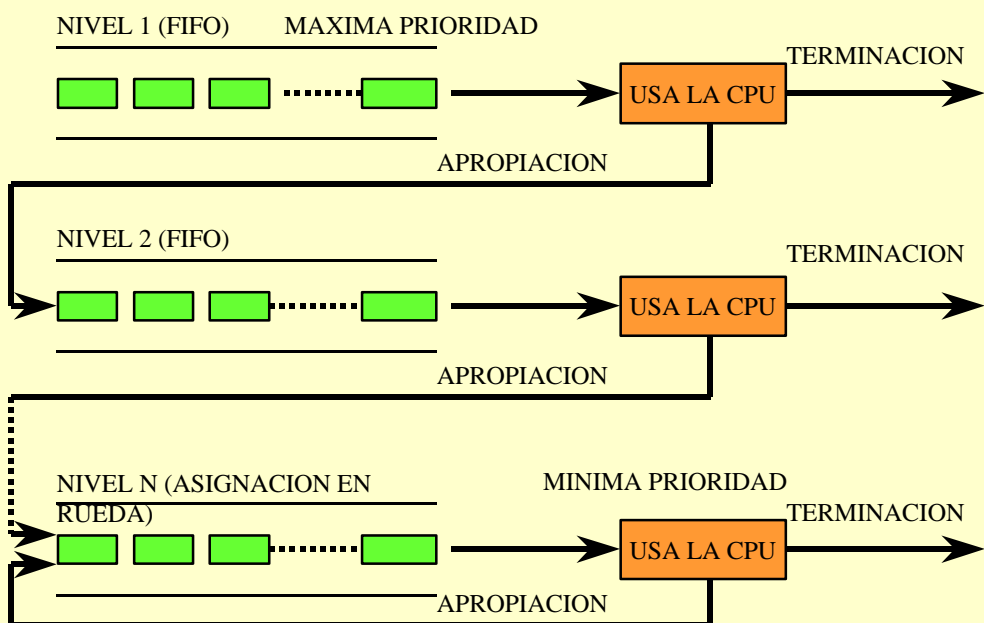
PLANIFICACION PRIMERO EN ENTRAR PRIMERO EN SALIR



PLANIFICACION DE ASIGNACION EN RUEDA (ROUND ROBIN: RR)



COLAS DE RETROALIMENTACION DE NIVELES MÚLTIPLES



TIPOS DE PLANIFICACION

- v **POLITICA VERSUS MECANISMO DE PLANIFICACION:**
- v PUEDE OCURRIR QUE HAYA PROCESOS CON MUCHOS PROCESOS HIJOS EJECUTANDOSE BAJO SU CONTROL:
 - u EJ.: PROCESO EN UN DBMS CON PROCESOS HIJOS ATENDIENDO FUNCIONES ESPECIFICAS:
 - Φ EJ.: ANALISIS DE INTERROGANTES, ACCESO A DISCOS, ETC.
- v ES POSIBLE QUE EL PROCESO PRINCIPAL (PADRE) PUEDA IDENTIFICAR LA IMPORTANCIA (O CRITICIDAD) DE LOS SUS PROCESOS HIJOS:
 - u PERO LOS PLANIFICADORES ANALIZADOS NO ACEPTAN DATOS DE LOS PROCESOS DE USUARIO RELATIVOS A DECISIONES DE PLANIFICACION.
 - u SOLUCION: SEPARAR EL **MECANISMO DE PLANIFICACION** DE LA **POLITICA DE PLANIFICACION**:
 - Φ SE PARAMETRIZA EL ALGORITMO DE PLANIFICACION.
 - Φ LOS PARAMETROS PUEDEN SER DETERMINADOS POR MEDIO DE PROCESOS DEL USUARIO.
 - Φ EL MECANISMO ESTA EN EL NUCLEO PERO LA POLITICA QUEDA ESTABLECIDA POR UN PROCESO DEL USUARIO.
- v **PLANIFICACION DE DOS NIVELES:**
- v LOS ESQUEMAS ANALIZADOS HASTA AHORA SUPONEN QUE TODOS LOS PROCESOS EJECUTABLES ESTAN EN LA MEMORIA PRINCIPAL.
- v SI LA MEMORIA PRINCIPAL ES INSUFICIENTE:
 - u HABRA PROCESOS EJECUTABLES QUE SE MANTENGAN EN DISCO.
 - u HABRA IMPORTANTES IMPLICACIONES PARA LA PLANIFICACION:
 - Φ EL TIEMPO DE ALTERNANCIA ENTRE PROCESOS PARA TRAER Y PROCESAR UN PROCESO DEL DISCO ES CONSIDERABLEMENTE MAYOR QUE EL TIEMPO PARA UN PROCESO QUE YA ESTA EN LA MEMORIA PRINCIPAL.
 - Φ ES MAS EFICIENTE EL INTERCAMBIO DE LOS PROCESOS CON UN PLANIFICADOR DE DOS NIVELES.

TIPOS DE PLANIFICACION

- v ESQUEMA OPERATIVO DE UN PLANIFICADOR DE DOS NIVELES:
 - u SE CARGA EN LA MEMORIA PRINCIPAL CIERTO SUBCONJUNTO DE LOS PROCESOS EJECUTABLES.
 - u EL PLANIFICADOR SE RESTRINGE A ELLOS DURANTE CIERTO TIEMPO.
 - u PERIODICAMENTE SE LLAMA A UN PLANIFICADOR DE NIVEL SUPERIOR PARA:
 - Φ ELIMINAR DE LA MEMORIA LOS PROCESOS QUE HAYAN PERMANECIDO EN ELLA EL TIEMPO SUFICIENTE.
 - Φ CARGAR A MEMORIA LOS PROCESOS QUE HAYAN ESTADO EN DISCO DEMASIADO TIEMPO.
 - u EL PLANIFICADOR DE NIVEL INFERIOR SE RESTRINGE DE NUEVO A LOS PROCESOS EJECUTABLES QUE SE ENCUENTREN EN LA MEMORIA.
 - u EL PLANIFICADOR DE NIVEL SUPERIOR SE ENCARGA DE DESPLAZAR LOS PROCESOS DE MEMORIA A DISCO Y VICEVERSA.
- v CRITERIOS QUE PODRIA UTILIZAR EL PLANIFICADOR DE NIVEL SUPERIOR PARA TOMAR SUS DECISIONES:
 - u ¿ CUANTO TIEMPO HA TRANSCURRIDO DESDE EL ULTIMO INTERCAMBIO DEL PROCESO ?.
 - u ¿ CUANTO TIEMPO DE CPU HA UTILIZADO RECIENTEMENTE EL PROCESO ?.
 - u ¿ QUE TAN GRANDE ES EL PROCESO ? (GENERALMENTE LOS PROCESOS PEQUEÑOS NO CAUSAN TANTOS PROBLEMAS EN ESTE SENTIDO).
 - u ¿ QUE TAN ALTA ES LA PRIORIDAD DEL PROCESO ?.
- v EL PLANIFICADOR DE NIVEL SUPERIOR PODRIA UTILIZAR CUALQUIERA DE LOS METODOS DE PLANIFICACION ANALIZADOS.

MULTIPROCESAMIENTO

✓ **INTRODUCCION:**

- ✓ ES UNA TENDENCIA SIGNIFICATIVA EN EL CAMPO DE LA COMPUTACION.
- ✓ CONSISTE EN CONFIGURAR UN SISTEMA DE COMPUTACION CON VARIOS PROCESADORES.
- ✓ NO ES UN ENFOQUE NUEVO PERO SI POSEE GRANDES PERSPECTIVAS EN FUNCION DEL DESARROLLO DE LOS MICROPROCESADORES.
- ✓ SE PODRIAN CONCEBIR SISTEMAS CONSTRUIDOS POR CIENTOS O MILES DE MICROPROCESADORES.

✓ **CONFIABILIDAD:**

- ✓ SI UN PROCESADOR FALLA LOS RESTANTES CONTINUAN OPERANDO:
 - ✓ NO ES AUTOMATICO Y REQUIERE DE UN DISEÑO CUIDADOSO.
- ✓ UN PROCESADOR QUE FALLA HABRA DE INFORMARLO A LOS DEMAS DE ALGUNA MANERA PARA QUE SE HAGAN CARGO DE SU TRABAJO.
- ✓ LOS PROCESADORES EN FUNCIONAMIENTO DEBEN PODER DETECTAR EL FALLO DE UN PROCESADOR DETERMINADO.
- ✓ EL S. O. DEBE:
 - ✓ PERCIBIR QUE HA FALLADO UN PROCESADOR DETERMINADO:
 - ⊕ YA NO PODRA ASIGNARLO.
 - ✓ AJUSTAR SUS ESTRATEGIAS DE ASIGNACION DE RECURSOS PARA EVITAR LA SOBRECARGA DEL SISTEMA QUE ESTA DEGRADADO.

✓ **EXPLOTACION DEL PARALELISMO:**

- ✓ “LA MAYORIA DE LOS SISTEMAS DE MULTIPROCESAMIENTO TIENEN COMO META PRINCIPAL EL INCREMENTO DE LA CAPACIDAD DE EJECUCION”.
- ✓ LA PROGRAMACION SIGUE SIENDO ESENCIALMENTE SECUENCIAL Y GENERALMENTE NO SE EXPLOTA LA CONCURRENCIA.
- ✓ LAS PRINCIPALES RAZONES SON:
 - ✓ LAS PERSONAS PIENSAN EN FORMA SECUENCIAL.
 - ✓ NINGUN LENGUAJE HUMANO PROPORCIONA LA EXPRESION ADECUADA DE PARALELISMO:
 - ⊕ EXISTEN LENGUAJES DE COMPUTACION CON SOPORTE DE

MULTIPROCESAMIENTO

- v NI EL MULTIPROCESAMIENTO HA SIDO USADO CON AMPLITUD PARA EXPLOTAR EL PARALELISMO.
- v EL HARDWARE TRADICIONAL DEL COMPUTADOR ESTA ORIENTADO HACIA LA OPERACION SECUENCIAL.
- v ES MUY DIFICIL DEPURAR PROGRAMAS EN PARALELO.
- v LOS MULTIPROCESADORES NO SE UTILIZAN A MENUDO PARA EXPLOTAR EL PARALELISMO:
 - v ES MUY ESCASO ES SOFTWARE QUE EXPLOTE EL PARALELISMO.
- v LO DESEABLE ES QUE LOS S. O. Y COMPILADORES PUEDAN DETECTAR E IMPLEMENTAR EL PARALELISMO AUTOMATICAMENTE.
- v **PARALELISMO MASIVO:**
- v SE DEBE DISPONER DE SUFICIENTES PROCESADORES COMO PARA QUE TODAS LAS OPERACIONES QUE PUEDAN SER EJECUTADAS EN PARALELO PUEDAN SER ASIGNADAS A PROCESADORES SEPARADOS.
- v OFRECE UNA FORMA DE EJECUTAR UN PROGRAMA EN EL MENOR TIEMPO POSIBLE.
- v LA CUESTION CENTRAL ES:
 - v DISPONIENDO DEL PARALELISMO MASIVO:
 - Φ ¿ CUAL ES EL TIEMPO MINIMO REQUERIDO PARA EJECUTAR UN ALGORITMO DETERMINADO ?.
- v **METAS DE LOS SISTEMAS DE MULTIPROCESAMIENTO:**
- v GENERALMENTE SON:
 - v CONFIABILIDAD Y DISPONIBILIDAD MUY ALTAS.
 - v INCREMENTO DEL PODER DE COMPUTACION.
- v SU DISEÑO MODULAR:
 - v PROPORCIONA UNA FLEXIBILIDAD IMPORTANTE.
 - v FACILITA LA EXPANSION DE LA CAPACIDAD.
- v **DETECCION AUTOMATICA DEL PARALELISMO:**
- v LOS MULTIPROCESADORES HACEN POSIBLE LA EXPLOTACION DEL PARALELISMO.
- v LOS SISTEMAS DE COMPUTACION OBTIENEN LOS BENEFICIOS DEL PROCESAMIENTO CONCURRENTES:

MULTIPROCESAMIENTO

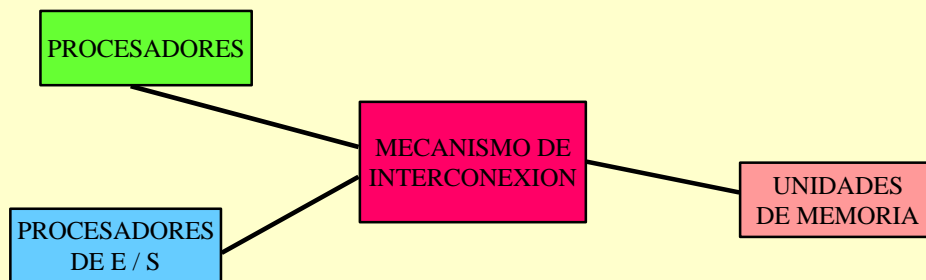
- v MAS POR LA “MULTIPROGRAMACION” DE VARIOS PROCESOS.
- v MENOS POR LA EXPLOTACION DEL “PARALELISMO” DENTRO DE UN SOLO PROCESO.
- v LA DETECCION DEL PARALELISMO:
 - v ES UN PROBLEMA COMPLEJO.
 - v LA PUEDE EFECTUAR EL PROGRAMADOR, EL TRADUCTOR DEL LENGUAJE, EL HARDWARE O EL S. O.
- v EL PARALELISMO DENTRO DE LOS PROGRAMAS PUEDE SER “EXPLICITO” O “IMPLICITO”.
- v EL PARALELISMO “EXPLICITO”:
 - v ES INDICADO DE FORMA ESPECIFICA POR UN PROGRAMADOR MEDIANTE UNA “CONSTRUCCION DE CONCURRENCIA” COMO:
 - Φ COBEGIN;
 - PROPOSICION 1;
 -
 - PROPOSICION N;
 - Φ COEND;
 - v SE PUEDEN UTILIZAR PROCESADORES SEPARADOS PARA EJECUTAR C / U DE LAS PROPOSICIONES.
 - v ES SUSCEPTIBLE DE ERRORES DE PROGRAMACION DIFICILES DE DETECTAR Y DEPURAR.
 - v EL PROGRAMADOR PUEDE OMITIR TRATAR SITUACIONES DONDE SERIA APLICABLE EL PARALELISMO.
- v EL PARALELISMO “IMPLICITO”:
 - v LA VERDADERA ESPERANZA ESTA EN LA DETECCION AUTOMATICA DEL PARALELISMO IMPLICITO.
 - v ES EL PARALELISMO INTRINSECO DEL ALGORITMO PERO NO ESTABLECIDO EXPLICITAMENTE POR EL PROGRAMADOR.
 - v LOS COMPILADORES EXPLOTAN EL PARALELISMO IMPLICITO MEDIANTE LAS TECNICAS DE:
 - Φ “DISTRIBUCION DE CICLOS”.
 - Φ “REDUCCION DE LA ALTURA DEL ARBOL”.

MULTIPROCESAMIENTO

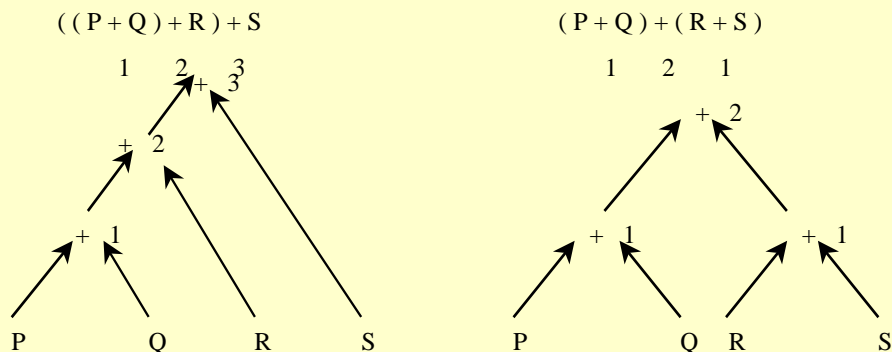
- v **DISTRIBUCION DE CICLOS:**
 - v UNA “ESTRUCTURA DE CICLOS O DE REPETICION” IMPLICA LA REPETICION DE UNA SERIE DE PROPOSICIONES (CUERPO DEL CICLO) HASTA QUE OCURRE ALGUNA CONDICION DE TERMINACION.
 - v EJ.: FOR I = 1 TO 3 DO A (I) = B (I) + C (I);
 - v EL PROCESADOR SECUENCIAL REALIZARA EN SECUENCIA:
 - v A (1) = B (1) + C (1); A (2) = B (2) + C (2); A (3) = B (3) + C (3);
 - v EN UN SISTEMA DE MULTIPROCESAMIENTO CON TRES PROCESADORES DISPONIBLES SE PODRIAN EJECUTAR CONCURRENTEMENTE.
 - v UN COMPILADOR QUE DETECTE AUTOMATICAMENTE EL PARALELISMO IMPLICITO PUEDE CONVERTIR EL CICLO DEL EJ. EN:
 - v COBEGIN;
 - v A (1) = B (1) + C (1); A (2) = B (2) + C (2); A (3) = B (3) + C (3);
 - v COEND;
 - v ESTA TECNICA SE DENOMINA “DISTRIBUCION DE CICLOS”.
 - v **REDUCCION DE LA ALTURA DEL ARBOL:**
 - v UTILIZANDO LAS PROPIEDADES ASOCIATIVA, CONMUTATIVA Y DISTRIBUTIVA DE LA ARITMETICA, LOS COMPILADORES PUEDEN:
 - v DETECTAR EL PARALELISMO IMPLICITO EN EXPRESIONES ALGEBRAICAS.
 - v PRODUCIR UN CODIGO OBJETO PARA MULTIPROCESADORES QUE INDIQUE LAS OPERACIONES QUE SE PUEDEN REALIZAR SIMULTANEAMENTE.
 - v REORDENAR EXPRESIONES PARA QUE SEAN MAS APROPIADAS PARA LA COMPUTACION EN PARALELO.
 - v SE INVIERTEN MAS TIEMPO Y RECURSOS DURANTE LA COMPILACION PARA REDUCIR EL TIEMPO DE EJECUCION:
 - v OPTIMIZACION EN EL MOMENTO DE LA COMPILACION PARA LOGRAR EJECUCION EN TIEMPO MINIMO:
 - Φ APLICABLE ESPECIALMENTE CUANDO LOS SISTEMAS PASAN A PRODUCCION, NO TANTO CUANDO ESTAN EN DESARROLLO.
- PROCESOS Y ADMINISTRACION DEL PROCESADOR 69

MULTIPROCESAMIENTO

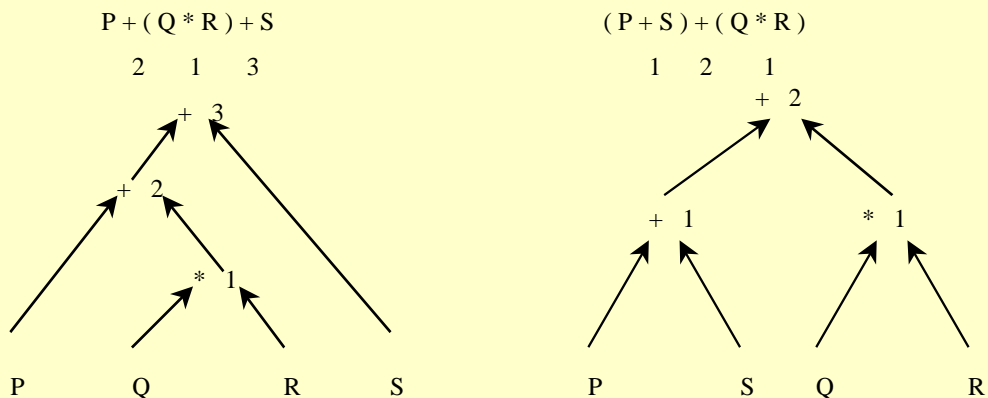
IDEA SIMPLIFICADA DE LA ORGANIZACION DE UN MULTIPROCESADOR



REDUCCION DE LA ALTURA DEL ARBOL POR ASOCIATIVIDAD

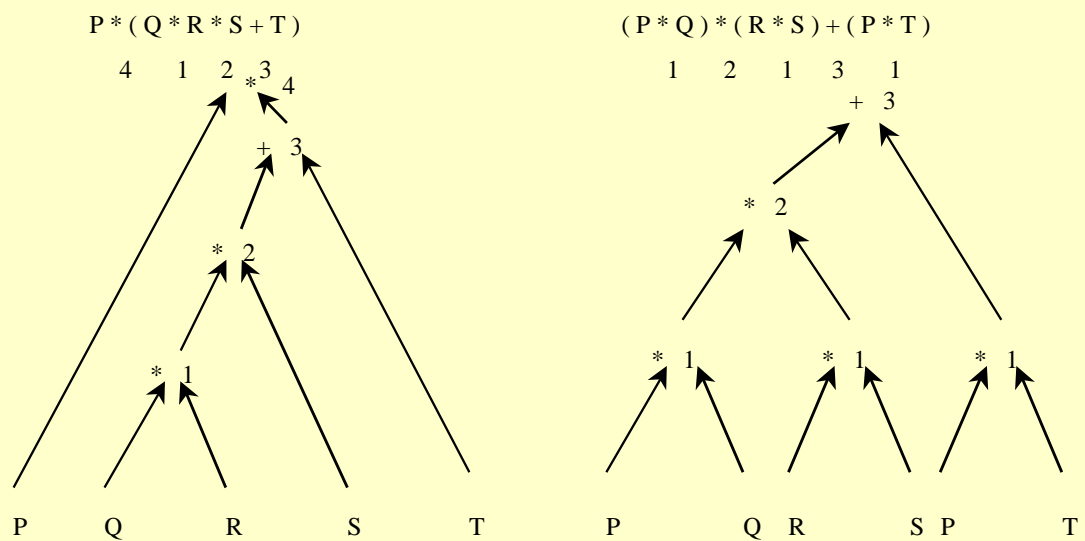


REDUCCION DE LA ALTURA DEL ARBOL POR CONMUTATIVIDAD



MULTIPROCESAMIENTO

REDUCCION DE LA ALTURA DEL ARBOL POR DISTRIBUTIVIDAD



REGLA DE “NUNCA ESPERAR”:

ES MEJOR DARLE A UN PROCESADOR UNA TAREA QUE PUEDE LLEGAR A NO SER UTILIZADA, QUE TENERLO OCIOSO.

ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR

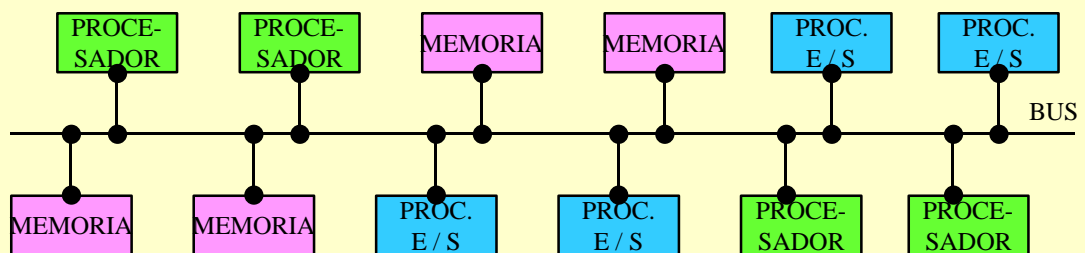
- v EL PROBLEMA CLAVE ES DETERMINAR LOS MEDIOS DE CONEXION DE LOS PROCESADORES MULTIPLES Y LOS PROCESADORES DE E / S A LAS UNIDADES DE ALMACENAMIENTO.
- v LOS MULTIPROCESADORES SE CARACTERIZAN POR:
 - u UN MULTIPROCESADOR CONTIENE DOS O MAS PROCESADORES CON CAPACIDADES APROXIMADAMENTE COMPARABLES.
 - u TODOS LOS PROCESADORES COMPARTEN EL ACCESO A:
 - Φ UN ALMACENAMIENTO COMUN.
 - Φ CANALES DE E / S, UNIDADES DE CONTROL Y DISPOSITIVOS.
 - u TODO ESTA CONTROLADO POR UN S. O. QUE PROPORCIONA INTERACCION ENTRE PROCESADORES Y SUS PROGRAMAS EN:
 - Φ LOS NIVELES DE TRABAJO, TAREA, PASO, ARCHIVO Y ELEMENTOS DE DATOS.
- v LAS ORGANIZACIONES MAS COMUNES SON:
 - u TIEMPO COMPARTIDO O BUS COMUN (CONDUCTOR COMUN).
 - u MATRIZ DE BARRAS CRUZADAS E INTERRUPTORES.
 - u ALMACENAMIENTO DE INTERCONEXION MULTIPLE.
- v **TIEMPO COMPARTIDO O BUS COMUN (O CONDUCTOR COMUN):**
- v USA UN SOLO CAMINO DE COMUNICACION ENTRE TODAS LAS UNIDADES FUNCIONALES.
- v EL BUS COMUN ES EN ESENCIA UNA UNIDAD PASIVA.
- v UN PROCESADOR O PROCESADOR DE E / S QUE DESEE TRANSFERIR DATOS DEBE:
 - u VERIFICAR LA DISPONIBILIDAD DEL CONDUCTOR Y DE LA UNIDAD DE DESTINO.
 - u INFORMAR A LA UNIDAD DE DESTINO DE LO QUE SE VA A HACER CON LOS DATOS.
 - u INICIAR LA TRANSFERENCIA DE DATOS.
- v LAS UNIDADES RECEPTORAS DEBEN PODER:
 - u RECONOCER QUE MENSAJES DEL BUS SON ENVIADOS HACIA ELLAS.
 - u SEGUIR Y CONFIRMAR LAS SEÑALES DE CONTROL RECIBIDAS DE LA UNIDAD EMISORA.

ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR

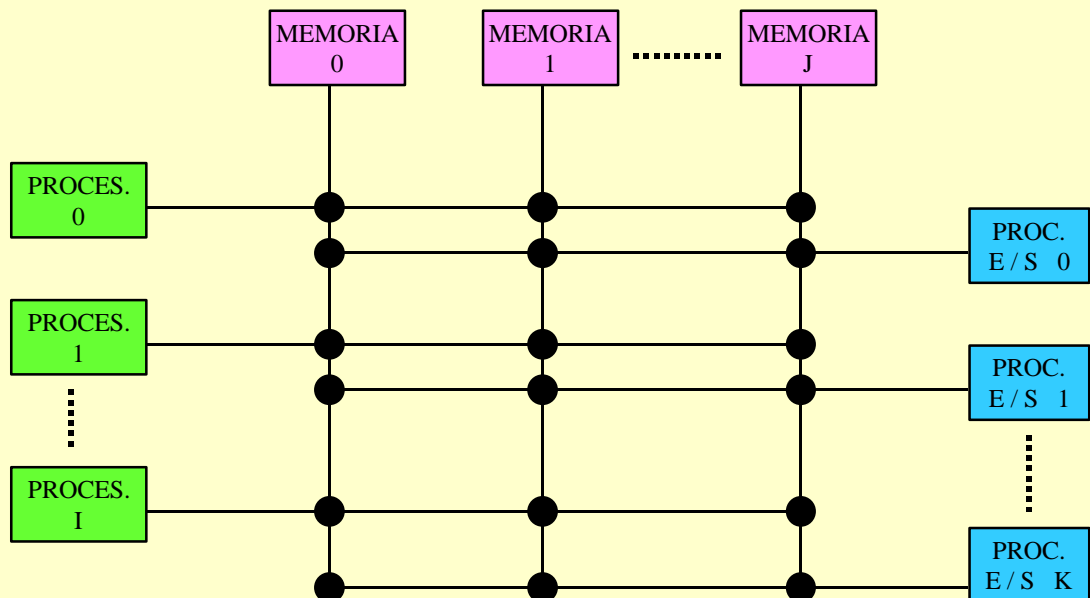
- v ES UNA ORGANIZACION ECONOMICA, SIMPLE Y FLEXIBLE PERO CON UNA SOLA VIA DE COMUNICACION:
 - u EL SISTEMA FALLA TOTALMENTE SI FALLA EL BUS.
 - u LA TASA NETA DE TRANSMISIONES ESTA LIMITADA POR LA TASA NETA DE TRANSMISION DEL CONDUCTOR.
 - u LA CONTENCIÓN POR EL USO DEL BUS EN UN SISTEMA SOBRECARGADO PUEDE OCASIONAR UNA SERIA DEGRADACION.
- v **MATRIZ DE BARRAS CRUZADAS E INTERRUPTORES:**
- v EXISTE UN CAMINO DIFERENTE PARA CADA UNIDAD DE ALMACENAMIENTO:
 - u LAS REFERENCIAS A DOS UNIDADES DIFERENTES DE ALMACENAMIENTO NO SON BLOQUEANTES SINO SIMULTANEAS.
 - u LA MULTIPLICIDAD DE CAMINOS DE TRANSMISION PUEDE PROPORCIONAR TASAS DE TRANSFERENCIA MUY ALTAS.
- v **ALMACENAMIENTO DE INTERCONEXION MULTIPLE:**
- v SE OBTIENE AL SACAR LAS LOGICAS DE CONTROL, DE CONMUTACION Y DE ARBITRAJE DE PRIORIDADES FUERA DEL INTERRUPTOR DE BARRAS CRUZADAS:
 - u SE LAS COLOCA EN LA INTERFAZ DE CADA UNIDAD DE ALMACENAMIENTO.
- v CADA UNIDAD FUNCIONAL PUEDE ACCEDER A CADA UNIDAD DE ALMACENAMIENTO, PERO SOLO EN UNA “CONEXION DE ALMACENAMIENTO” ESPECIFICA:
 - u HAY UNA CONEXION DE ALMACENAMIENTO POR UNIDAD FUNCIONAL.
- v EL CONEXIONADO ES MAS COMPLEJO QUE EN LOS OTROS ESQUEMAS.
- v SE PUEDE RESTRINGIR EL ACCESO A LAS UNIDADES DE ALMACENAMIENTO PARA QUE NO TODAS LAS UNIDADES DE PROCESAMIENTO LAS ACCEDAN:
 - u HABRA UNIDADES DE ALMACENAMIENTO “PRIVADAS” DE DETERMINADOS PROCESADORES.

ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR

ORGANIZACION DE MULTIPROCESADOR DE TIEMPO COMPARTIDO DE BUS COMUN

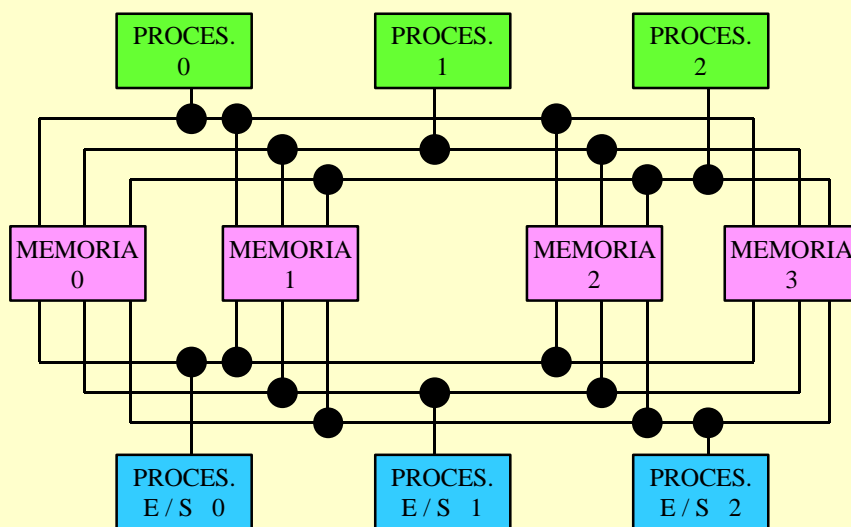


ORGANIZACION DEL MULTIPROCESADOR POR MATRIZ DE BARRAS CRUZADAS E INTERRUPTORES

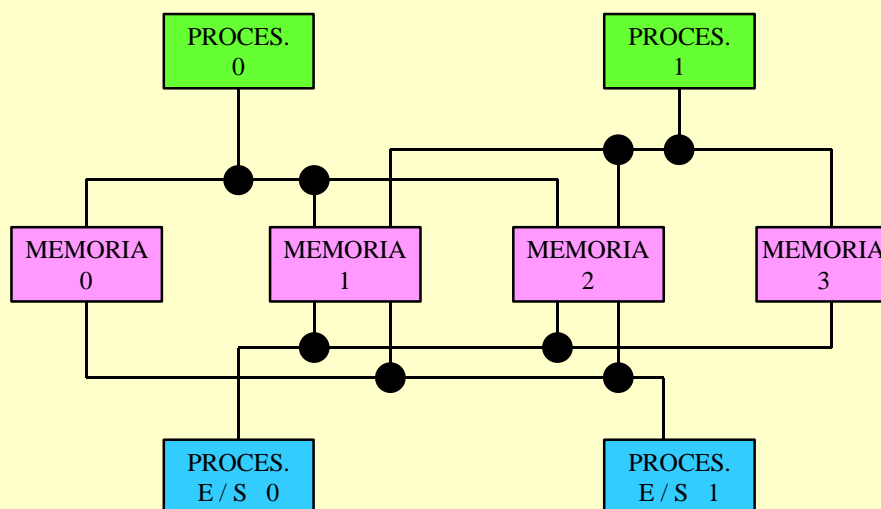


ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR

ORGANIZACION DE MULTIPROCESADOR POR SISTEMA DE MEMORIA DE INTERCONEXION MULTIPLE



ORGANIZACION DEL MULTIPROCESADOR POR SISTEMA DE MEMORIA DE INTERCONEXION MULTIPLE CON MEMORIAS PRIVADAS

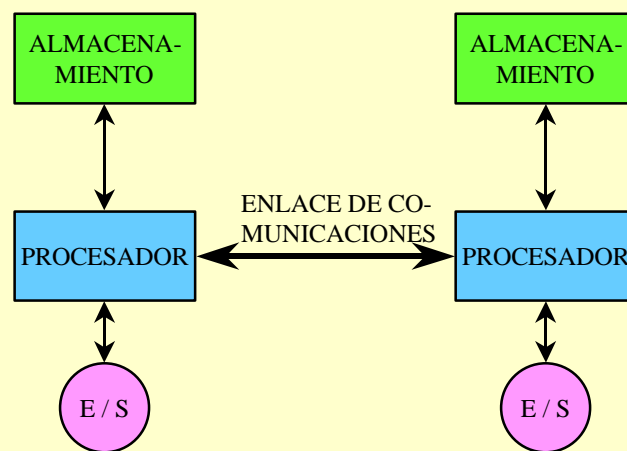


GRADOS DE ACOPLAMIENTO EN MULTIPROCESAMIENTO

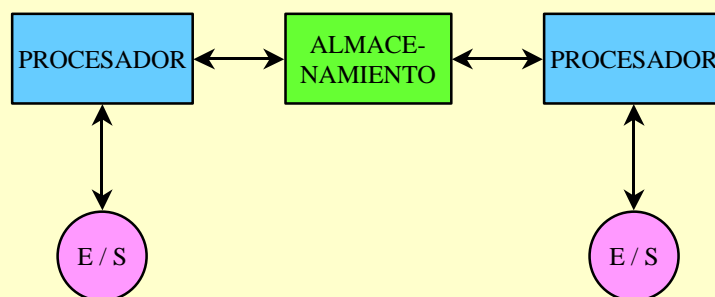
- v “MULTIPROCESAMIENTO LIGERAMENTE ACOPLADO”:
 - u INCLUYE LA CONEXION DE DOS O MAS SISTEMAS INDEPENDIENTES POR MEDIO DE UN ENLACE DE COMUNICACION.
 - u CADA SISTEMA TIENE SU PROPIO S. O. Y ALMACENAMIENTO.
 - u LOS SISTEMAS PUEDEN FUNCIONAR INDEPENDIENTEMENTE:
 - Φ SE COMUNICAN CUANDO SEA NECESARIO.
 - u LOS SISTEMAS SEPARADOS PUEDEN:
 - Φ ACCEDER A LOS ARCHIVOS DE LOS OTROS.
 - Φ INTERCAMBIAR TAREAS A PROCESADORES MENOS CARGADOS.
- v “MULTIPROCESAMIENTO RIGIDAMENTE ACOPLADO”:
 - u UTILIZA UN SOLO ALMACENAMIENTO COMPARTIDO POR VARIOS PROCESADORES.
 - u EMPLEA UN SOLO S. O. QUE CONTROLA TODOS LOS PROCESADORES Y EL HARDWARE DEL SISTEMA.
- v **ORGANIZACION MAESTRO / SATELITE:**
- v UN PROCESADOR ESTA DISEÑADO COMO EL “MAESTRO” Y LOS OTROS COMO “SATELITES”.
- v EL PROCESADOR MAESTRO ES DE PROPOSITO GRAL. Y REALIZA:
 - u OPERACIONES DE E / S Y COMPUTACIONES.
- v LOS PROCESADORES SATELITES SOLO REALIZAN COMPUTACIONES.
- v LOS PROCESOS LIMITADOS POR COMPUTACION PUEDEN EJECUTARSE CON EFECTIVIDAD EN LOS SATELITES.
- v LOS PROCESOS LIMITADOS POR LA E / S EJECUTADOS EN LOS SATELITES GENERAN FRECUENTES LLAMADAS DE SERVICIOS AL PROCESADOR MAESTRO, PUDIENDO RESULTAR INEFICIENTES.
- v SI FALLA UN SATELITE SE PIERDE CAPACIDAD COMPUTACIONAL PERO EL SISTEMA NO FALLA.
- v SI FALLA EL MAESTRO EL SISTEMA FALLA AL NO PODER EFECTUAR OPERACIONES DE E / S:
 - u UN SATELITE DEBERIA ASUMIR LAS FUNCIONES DEL MAESTRO PREVIO CAMBIO DE LOS PERIFERICOS Y REINICIO DEL SISTEMA.

GRADOS DE ACOPLAMIENTO EN MULTIPROCESAMIENTO

MULTIPROCESAMIENTO LIGERAMENTE ACOPLADO



MULTIPROCESAMIENTO RIGIDAMENTE ACOPLADO



S. O. DE MULTIPROCESADORES

- v LAS CAPACIDADES FUNCIONALES DE LOS S. O. DE MULTIPROGRAMACION Y DE MULTIPROCESADORES INCLUYEN:
 - υ ASIGNACION Y ADMINISTRACION DE RECURSOS.
 - υ PROTECCION DE TABLAS Y CONJUNTOS DE DATOS.
 - υ PREVENCIÓN CONTRA EL INTERBLOQUEO DEL SISTEMA.
 - υ TERMINACION ANORMAL.
 - υ EQUILIBRIO DE CARGAS DE ENTRADA / SALIDA.
 - υ EQUILIBRIO DE CARGA DEL PROCESADOR.
 - υ RECONFIGURACION.
- v LAS TRES ULTIMAS SON ESPECIALMENTE IMPORTANTES EN S. O. DE MULTIPROCESADORES:
 - υ ES FUNDAMENTAL:
 - Φ EXPLOTAR EL PARALELISMO EN EL HARDWARE Y EN LOS PROGRAMAS.
 - Φ HACERLO AUTOMATICAMENTE.
- v LAS ORGANIZACIONES BASICAS DE LOS S. O. PARA MULTIPROCESADORES SON:
 - υ MAESTRO / SATELITE.
 - υ EJECUTIVO SEPARADO PARA CADA PROCESADOR.
 - υ TRATAMIENTO SIMETRICO (O ANONIMO) PARA TODOS LOS PROCESADORES.
- v **MAESTRO SATELITE:**
- v ES LA ORGANIZACION MAS FACIL DE IMPLEMENTAR.
- v NO LOGRA LA UTILIZACION OPTIMA DEL HARDWARE:
 - υ SOLO EL PROCESADOR MAESTRO PUEDE EJECUTAR EL S. O.
 - υ EL PROCESADOR SATELITE SOLO PUEDE EJECUTAR PROGRAMAS DEL USUARIO.
- v LAS INTERRUPCIONES GENERADAS POR LOS PROCESOS EN EJECUCION EN LOS PROCESADORES SATELITES QUE PRECISAN ATENCION DEL S. O.:
 - υ DEBEN SER ATENDIDAS POR EL PROCESADOR MAESTRO.
 - υ PUEDEN GENERARSE LARGAS COLAS DE REQUERIMIENTOS PENDIENTES.

S. O. DE MULTIPROCESADORES

- v **EJECUTIVOS SEPARADOS:**
- v CADA PROCESADOR:
 - v TIENE SU PROPIO S. O.
 - v RESPONDE A INTERRUPCIONES DE LOS USUARIOS QUE OPERAN EN ESE PROCESADOR.
- v EXISTEN TABLAS DE CONTROL CON INFORMACION GLOBAL DE TODO EL SISTEMA (EJ.: LISTA DE PROCESADORES CONOCIDOS POR EL S. O.):
 - v SE LAS DEBE ACCEDER UTILIZANDO EXCLUSION MUTUA.
- v ES MAS CONFIABLE QUE LA ORGANIZACION MAESTRO / SATELITE.
- v CADA PROCESADOR CONTROLA SUS PROPIOS RECURSOS DEDICADOS.
- v LA RECONFIGURACION DE LOS DISPOSITIVOS DE E / S PUEDE:
 - v IMPLICAR EL CAMBIO DE DISPOSITIVOS A DIFERENTES PROCESADORES CON DISTINTOS S. O.
- v LA CONTENCIÓN SOBRE LAS TABLAS DEL S. O. ES MINIMA.
- v LOS PROCESADORES NO COOPERAN EN LA EJECUCION DE UN PROCESO INDIVIDUAL, QUE HABRA SIDO ASIGNADO A UNO DE ELLOS.
- v **TRATAMIENTO SIMETRICO:**
- v ES LA ORGANIZACION MAS COMPLICADA DE IMPLEMENTAR Y TAMBIEN LA MAS PODEROSA Y CONFIABLE.
- v EL S. O. ADMINISTRA UN GRUPO DE PROCESADORES IDENTICOS:
 - v CUALQUIERA PUEDE UTILIZAR CUALQUIER DISPOSITIVO DE E / S.
 - v CUALQUIERA PUEDE REFERENCIAR A CUALQUIER UNIDAD DE ALMACENAMIENTO.
- v EL S. O. PRECISA CODIGO REENTRANTE Y EXCLUSION MUTUA.
- v ES POSIBLE EQUILIBRAR LA CARGA DE TRABAJO MAS PRECISAMENTE QUE EN LAS OTRAS ORGANIZACIONES.
- v ADQUIEREN SIGNIFICATIVA IMPORTANCIA EL HARDWARE Y EL SOFTWARE PARA RESOLUCION DE CONFLICTOS.
- v TODOS LOS PROCESADORES PUEDEN COOPERAR EN LA EJECUCION DE UN PROCESO DETERMINADO.
- v “PROCESADOR EJECUTIVO”: ES EL RESPONSABLE (UNO SOLO) EN UN MOMENTO DADO DE LAS TABLAS Y FUNCIONES DEL SISTEMA:

RENDIMIENTO DEL SISTEMA DE MULTIPROCESAMIENTO

RECUPERACION DE ERRORES

- v AUN CON MULTIPROCESAMIENTO COMPLETAMENTE SIMETRICO LA ADICION DE UN NUEVO PROCESADOR NO HARA QUE LA CAPACIDAD DE EJECUCION DEL SISTEMA AUMENTE SEGUN LA CAPACIDAD DEL NUEVO PROCESADOR:
 - v HAY SOBRECARGA ADICIONAL DEL S. O.
 - v SE INCREMENTA LA CONTENCION POR RECURSOS DEL SISTEMA.
 - v HAY RETRASOS DEL HARDWARE EN:
 - Φ EL INTERCAMBIO.
 - Φ EL ENCAMINAMIENTO DE LAS TRANSMISIONES ENTRE UN N° MAYOR DE COMPONENTES.
- v AL INCREMENTAR EL N° DE PROCESADORES “N” SIMILARES EN UN MULTIPROCESADOR, EL INCREMENTO DE LA PRODUCTIVIDAD:
 - v NO ES LINEAL.
 - v TIENDE A DISMINUIR CUANDO “N” CRECE.
- v UNA DE LAS CAPACIDADES MAS IMPORTANTES DE LOS S. O. DE MULTIPROCESADORES ES LA DE SOPORTAR FALLAS DE HARDWARE EN PROCESADORES INDIVIDUALES Y CONTINUAR SU OPERACION.
- v DEBE EXISTIR EL SOPORTE CORRESPONDIENTE EN EL S. O.
- v LAS TECNICAS DE RECUPERACION DE ERRORES INCLUYEN:
 - v LOS DATOS CRITICOS (DEL SISTEMA Y DE USUARIO) DEBEN MANTENERSE EN COPIAS MULTIPLES Y EN BANCOS DE ALMACENAMIENTO SEPARADOS.
 - v EL S. O. DEBE EJECUTAR EFECTIVAMENTE CON LA CONFIGURACION MAXIMA Y CON SUBCONJUNTOS ANTE FALLAS.
 - v DEBE HABER CAPACIDAD DE DETECCION Y CORRECCION DE ERRORES DE HARDWARE SIN INTERFERIR CON LA EFICIENCIA OPERACIONAL DEL SISTEMA.
 - v SE DEBE UTILIZAR LA CAPACIDAD OCIOSA DEL PROCESADOR PARA TRATAR DE DETECTAR POSIBLES FALLOS ANTES DE QUE SE PRODUZCAN.
 - v EL S. O. DEBE DIRIGIR UN PROCESADOR OPERATIVO PARA QUE TOME EL CONTROL DE UN PROCESO QUE SE ESTABA EJECUTANDO

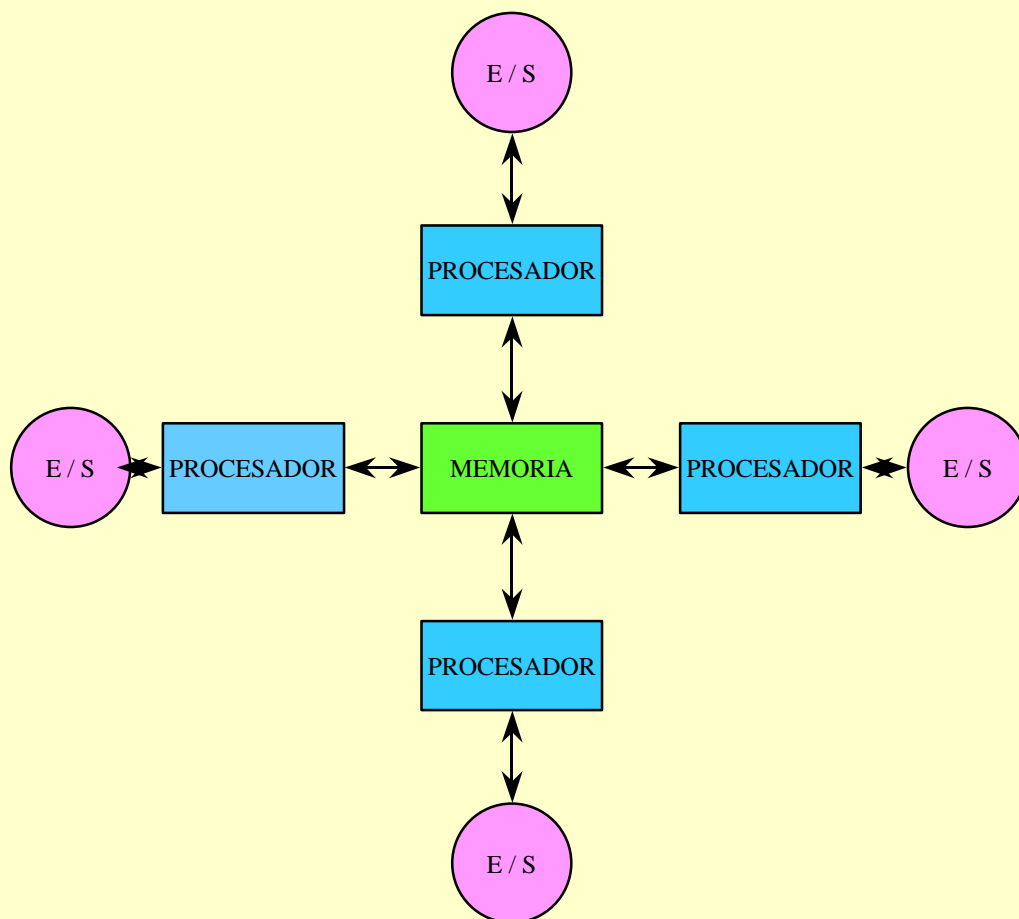
MULTIPROCESAMIENTO SIMETRICO

(MPS)

- v CADA PROCESADOR POSEE CAPACIDADES FUNCIONALES COMPLETAS.
- v LOS DISPOSITIVOS DE E / S PUEDEN SER CONECTADOS A CADA UNO DE LOS PROCESADORES.
- v TODAS LAS LLAMADAS AL SUPERVISOR PUEDEN SER EJECUTADAS EN TODOS LOS PROCESADORES:
 - u INCLUSIVE LAS DE E / S.
- v SI UN PROGRAMA EN EJECUCION EN UN PROCESADOR PIDE UNA OPERACION DE E / S EN UN DISPOSITIVO CONECTADO A UN PROCESADOR DIFERENTE:
 - u EL PROCESADOR PUEDE CONTINUAR EJECUTANDO EL TRABAJO.
 - u LA E / S SE COLOCA EN UNA COLA PARA SU INICIACION POR EL PROCESADOR APROPIADO.
- v SE CONSIDERA “PROCESADOR EJECUTANTE” AL QUE ESTA EJECUTANDO UN PROCESO DETERMINADO.
- v SE CONSIDERA “PROCESADOR PROPIETARIO” AL QUE ESTA CONECTADO A LOS DIFERENTES DISPOSITIVOS UTILIZADOS POR EL PROCESO.
- v ES MAS EFICIENTE QUE LA ORGANIZACION MAESTRO / SATELITE:
 - u LOS REQUERIMIENTOS DE E / S SE ENCOLAN Y NO SOBRECARGAN CON INTERCAMBIO DE CONTEXTO.
 - u EN LA ORGANIZACION MAESTRO / SATELITE LAS PETICIONES DE E /S EN EL SATELITE PROVOCAN UN INTERCAMBIO DE CONTEXTO EN EL MAESTRO.
- v CADA PROCESADOR PUEDE EJECUTAR EL PLANIFICADOR PARA BUSCAR EL SIGUIENTE TRABAJO A EJECUTAR:
 - u UN PROCESO DETERMINADO SE EJECUTA EN DIFERENTES PROCESADORES EN DISTINTOS MOMENTOS.
 - u MPS UTILIZA UNA SOLA COLA DE TRABAJOS Y CADA PROCESADOR PUEDE SELECCIONAR TRABAJOS DE ELLA:
 - ⊕ SE EQUILIBRA LA CARGA ENTRE LOS PROCESADORES.
 - ⊕ PARA MINIMIZAR LA CONTENCION EN EL DESPACHO DE PROCESOS LOS RELOJES DE LOS PROCESADORES TIENEN OBLICUIDAD:
 - LAS INTERRUPCIONES DE RELOJ OCURREN EN DIFERENTES MOMENTOS.

MULTIPROCESAMIENTO SIMETRICO (MPS)

EJEMPLO DE IMPLEMENTACION DE MULTIPROCESAMIENTO SIMETRICO



TENDENCIAS DE LOS MULTIPROCESADORES

- v TODO INDICA QUE EL USO DE LOS MULTIPROCESADORES SE INCREMENTARA CONSIDERABLEMENTE EN EL FUTURO.
- v LAS PRINCIPALES RAZONES SON:
 - u LA CONFIABILIDAD REQUERIDA ES CADA VEZ MAYOR.
 - u LA REDUCCION DE COSTOS CONSECUENCIA DE LOS AVANCES EN MICROELECTRONICA.
 - u EL PREVISIBLE DESARROLLO DE LENGUAJES QUE PERMITAN A LOS USUARIOS EXPRESAR EL PARALELISMO EXPLICITAMENTE.
 - u EL PROGRESO EN LA DETECCION AUTOMATICA DEL PARALELISMO.
 - u EL HECHO DE QUE SE ESTARIA LLEGANDO A LOS LIMITES DEL UNIPROCESADOR DEBIDO A LA COMPACTACION DE COMPONENTES:
 - Φ SE ESTARIA PROXIMO A LOS LIMITES DE LONGITUD Y DE PROXIMIDAD DE LOS “CAMINOS ELECTROMAGNETICOS”:
 - LONGITUD DEL RECORRIDO DE LA SEÑAL ELECTROMAGNETICA.
 - Φ ALCANZADOS LOS LIMITES MENCIONADOS LA UNICA POSIBILIDAD DE INCREMENTAR CAPACIDAD DE COMPUTO ES MEDIANTE MULTIPROCESAMIENTO.
- v EXISTEN ESTUDIOS DE TENDENCIAS EN ARQUITECTURA DE COMPUTADORAS QUE APUNTAR A LOS “POLIPROCESADORES”:
 - u SISTEMAS QUE COMBINAN EL MULTIPROCESAMIENTO, SIMETRICO Y ASIMETRICO, PARA CREAR UNA JERARQUIA DE PROCESADORES DENTRO DE UN SISTEMA.

ADMINISTRACION DE LA MEMORIA

- v **ALMACENAMIENTO REAL:**
- v INTRODUCCION
- v ORGANIZACION Y ADMINISTRACION DEL ALMACENAMIENTO
- v JERARQUIA DE ALMACENAMIENTO
- v ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO
- v MULTIPROGRAMACION DE PARTICION FIJA
- v MULTIPROGRAMACION DE PARTICION VARIABLE
- v MULTIPROGRAMACION CON INTERCAMBIO DE ALMACENAMIENTO
- v **ORGANIZACION DEL ALMACENAMIENTO VIRTUAL:**
- v INTRODUCCION
- v CONCEPTOS BASICOS DE ALMACENAMIENTO VIRTUAL
- v ORGANIZACION DEL ALMACENAMIENTO DE NIVELES MULTIPLES
- v TRANSFORMACION DE BLOQUES
- v CONCEPTOS BASICOS DE PAGINACION
- v SEGMENTACION
- v SISTEMAS DE PAGINACION / SEGMENTACION
- v **ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL:**
- v INTRODUCCION
- v ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL
- v LOCALIDAD
- v CONJUNTOS DE TRABAJO
- v PAGINACION POR DEMANDA Y PAGINACION ANTICIPADA
- v LIBERACION DE PAGINA Y TAMAÑO DE PAGINA
- v COMPORTAMIENTO DE UN PROGRAMA EN LA PAGINACION

INTRODUCCION

- v LA ORGANIZACION Y ADMINISTRACION DE LA “MEMORIA PRINCIPAL”, “MEMORIA PRIMARIA” O “MEMORIA REAL” DE UN SISTEMA:
 - u HA SIDO Y ES UNO DE LOS FACTORES MAS IMPORTANTES EN EL DISEÑO DE LOS S. O.
- v LOS TERMINOS “MEMORIA” Y “ALMACENAMIENTO” SE CONSIDERAN EQUIVALENTES.
- v LOS PROGRAMAS Y DATOS DEBEN ESTAR EN EL ALMACENAMIENTO PRINCIPAL PARA:
 - u PODERLOS EJECUTAR.
 - u REFERENCIARLOS DIRECTAMENTE.
- v SE CONSIDERA “ALMACENAMIENTO SECUNDARIO” O “ALMACENAMIENTO AUXILIAR” AL GENERALMENTE SOPORTADO EN DISCOS.
- v LOS HECHOS DEMUESTRAN QUE GENERALMENTE LOS PROGRAMAS CRECEN EN REQUERIMIENTOS DE MEMORIA TAN RAPIDO COMO LAS MEMORIAS:
 - u “LEY DE PARKINSON PARAFRASEADA”:
 - Φ LOS PROGRAMAS SE DESARROLLAN PARA OCUPAR TODA LA MEMORIA DISPONIBLE PARA ELLOS.
- v LA PARTE DEL S. O. QUE ADMINISTRA LA MEMORIA SE LLAMA **ADMINISTRADOR DE LA MEMORIA**:
 - u LLEVA UN REGISTRO DE LAS PARTES DE MEMORIA QUE SE ESTAN UTILIZANDO Y DE AQUELLAS QUE NO.
 - u ASIGNA ESPACIO EN MEMORIA A LOS PROCESOS CUANDO ESTOS LA NECESITAN.
 - u LIBERA ESPACIO DE MEMORIA ASIGNADA A PROCESOS QUE HAN TERMINADO.

ORGANIZACION Y ADMINISTRACION DEL ALMACENAMIENTO

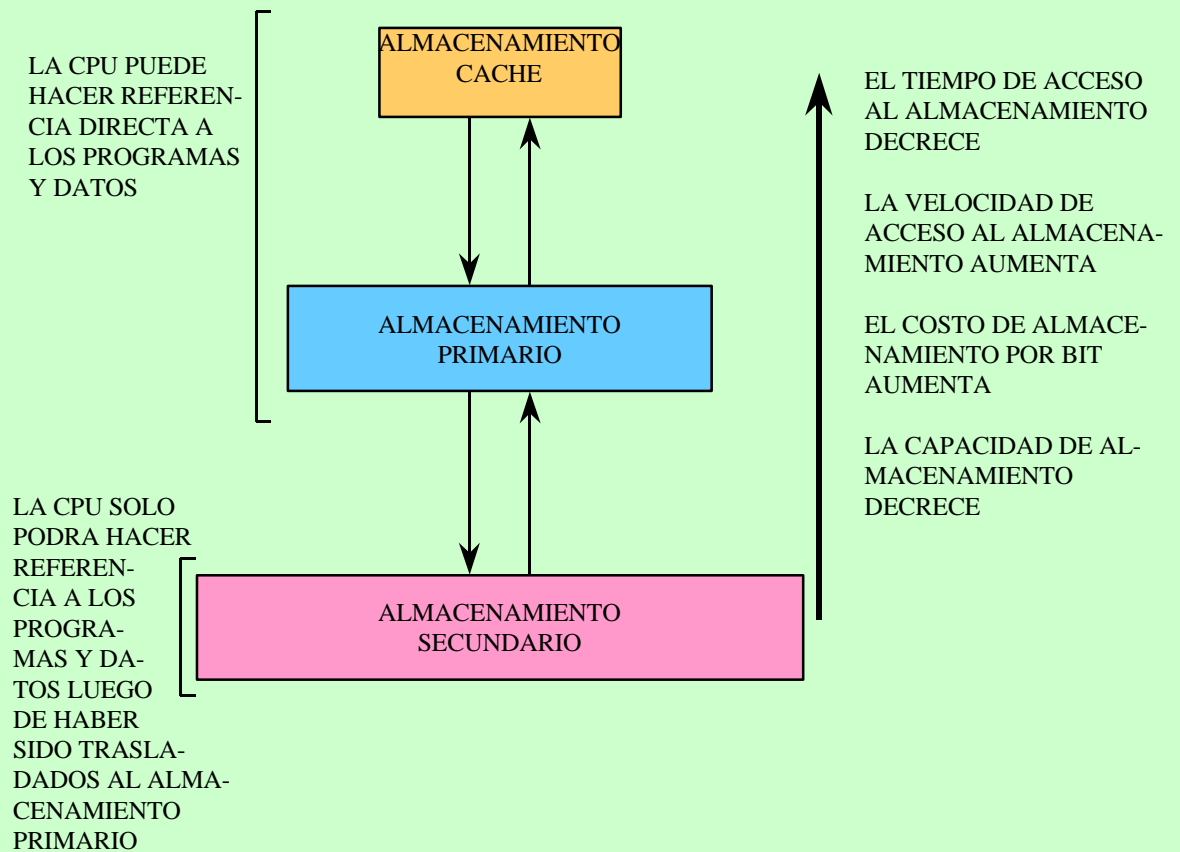
- v **ORGANIZACION DEL ALMACENAMIENTO:**
- v HISTORICAMENTE EL ALMACENAMIENTO PRINCIPAL SE HA CONSIDERADO COMO UN RECURSO COSTOSO:
 - u SU UTILIZACION DEBIA OPTIMIZARSE.
- v POR ORGANIZACION DEL ALMACENAMIENTO SE ENTIENDE LA MANERA DE CONSIDERAR ESTE ALMACENAMIENTO:
 - u ¿ SE COLOCA UN SOLO PROGRAMA DE USUARIO O VARIOS ?.
 - u SI SE ENCUENTRAN VARIOS PROGRAMAS DE USUARIO:
 - Φ ¿ SE CONCEDE A C / U LA MISMA CANTIDAD DE ESPACIO O SE DIVIDE EL ALMACENAMIENTO EN PORCIONES O “PARTICIONES” DE DIFERENTE TAMAÑO ?.
 - Φ ¿ SE UTILIZARA UN ESQUEMA RIGIDO DE NUMERO Y TAMAÑO DE PARTICIONES O UN ESQUEMA DINAMICO Y ADAPTABLE ?.
 - Φ ¿ SE REQUERIRA QUE LOS TRABAJOS DE LOS USUARIOS SEAN DISEÑADOS PARA FUNCIONAR EN UNA PARTICION ESPECIFICA O SE PERMITIRA QUE SE EJECUTEN EN CUALQUIERA DONDE QUEPAN ?.
 - Φ ¿ SE REQUERIRA O NO QUE CADA TRABAJO SEA COLOCADO EN UN BLOQUE CONTIGUO DE MEMORIA ?.
- v **ADMINISTRACION DEL ALMACENAMIENTO:**
- v INDEPENDIENTEMENTE DEL ESQUEMA DE ORGANIZACION HAY QUE DECIDIR LAS ESTRATEGIAS QUE SE UTILIZARAN PARA OPTIMIZAR EL RENDIMIENTO.
- v LAS ESTRATEGIAS DE ADMINISTRACION DEBEN CONSIDERAR:
 - u ¿ CUANDO SE CONSIGUE UN NUEVO PROGRAMA PARA COLOCAR EN LA MEMORIA ?:
 - Φ ¿ CUANDO EL SISTEMA LO PIDE ESPECIFICAMENTE O SE INTENTA ANTICIPARSE A LAS PETICIONES ?.
 - u ¿ DONDE SE COLOCARA EL PROGRAMA QUE SE EJECUTARA A CONTINUACION ?:
 - Φ ¿ SE PRIORIZA EL TIEMPO DE CARGA O LA OPTIMIZACION EN EL USO DEL ALMACENAMIENTO ?.
 - u ¿ CON QUE CRITERIO SE DESPLAZARAN PROGRAMAS ?.

JERARQUIA DE ALMACENAMIENTO

- v LOS PROGRAMAS Y DATOS TIENEN QUE ESTAR EN LA MEMORIA PRINCIPAL PARA PODER EJECUTARSE O SER REFERENCIADOS.
- v LOS PROGRAMAS Y DATOS QUE NO SON NECESARIOS DE INMEDIATO PUEDEN MANTENERSE EN EL ALMACENAMIENTO SECUNDARIO.
- v EL ALMACENAMIENTO PRINCIPAL ES MAS COSTOSO Y MENOR QUE EL SECUNDARIO PERO DE ACCESO MAS RAPIDO.
- v LOS SISTEMAS CON VARIOS NIVELES DE ALMACENAMIENTO REQUIEREN DESTINAR RECURSOS PARA ADMINISTRAR EL MOVIMIENTO DE PROGRAMAS Y DATOS ENTRE NIVELES.
- v UN NIVEL ADICIONAL ES EL “CACHE” O MEMORIA DE ALTA VELOCIDAD:
 - v ES MAS RAPIDA Y COSTOSA QUE LA MEMORIA PRINCIPAL.
 - v IMPONE AL SISTEMA UN NIVEL MAS DE TRASPASO:
 - Φ LOS PROGRAMAS SON TRASPASADOS DE LA MEMORIA PRINCIPAL AL CACHE ANTES DE SU EJECUCION.
 - v LOS PROGRAMAS EN LA MEMORIA CACHE EJECUTAN MUCHO MAS RAPIDO QUE EN LA MEMORIA PRINCIPAL.
 - v AL UTILIZAR MEMORIA CACHE SE ESPERA QUE:
 - Φ LA SOBRECARGA QUE SUPONE EL TRASPASO DE PROGRAMAS DE UN NIVEL DE MEMORIA A OTRO SEA MUCHO MENOR QUE,
 - Φ LA MEJORA EN EL RENDIMIENTO OBTENIDA POR LA POSIBILIDAD DE UNA EJECUCION MUCHO MAS RAPIDA EN LA CACHE.

JERARQUIA DE ALMACENAMIENTO

ORGANIZACION JERARQUICA DEL ALMACENAMIENTO



ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO

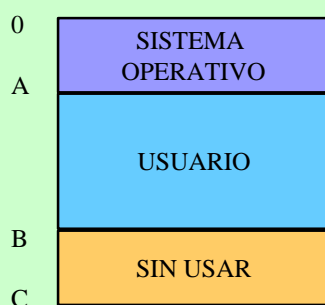
- v ESTAN DIRIGIDAS A LA OBTENCION DEL MEJOR USO POSIBLE DEL RECURSO DEL ALMACENAMIENTO PRINCIPAL.
- v SE DIVIDEN EN LAS SIGUIENTES CATEGORIAS:
 - u ESTRATEGIAS DE BUSQUEDA:
 - Φ ESTRATEGIAS DE BUSQUEDA POR DEMANDA.
 - Φ ESTRATEGIAS DE BUSQUEDA ANTICIPADA.
 - u ESTRATEGIAS DE COLOCACION.
 - u ESTRATEGIAS DE REPOSICION.
- v LAS “ESTRATEGIAS DE BUSQUEDA” ESTAN RELACIONADAS CON EL HECHO DE CUANDO OBTENER EL SIGUIENTE FRAGMENTO DE PROGRAMA O DE DATOS PARA SU INSERCIÓN EN LA MEMORIA PRINCIPAL.
- v EN LA “BUSQUEDA POR DEMANDA” EL SIGUIENTE FRAGMENTO DE PROGRAMA O DE DATOS SE CARGA AL ALMACENAMIENTO PRINCIPAL CUANDO ALGUN PROGRAMA EN EJECUCIÓN LO REFERENCIA.
- v SE CONSIDERA QUE LA “BUSQUEDA ANTICIPADA” PUEDE PRODUCIR UN MEJOR RENDIMIENTO DEL SISTEMA.
- v LAS “ESTRATEGIAS DE COLOCACION” ESTAN RELACIONADAS CON LA DETERMINACION DEL LUGAR DE LA MEMORIA DONDE SE COLOCARA (CARGARA) UN PROGRAMA NUEVO.
- v LAS “ESTRATEGIAS DE REPOSICION” ESTAN RELACIONADAS CON LA DETERMINACION DE QUE FRAGMENTO DE PROGRAMA O DE DATOS DESPLAZAR:
 - u PARA DAR LUGAR A LOS PROGRAMAS NUEVOS.
- v **ASIGNACION CONTIGUA DE ALMACENAMIENTO VERSUS NO CONTIGUA:**
- v EN LA “ASIGNACION CONTIGUA” CADA PROGRAMA OCUPA UN BLOQUE CONTIGUO Y SENCILLO DE LOCALIZACIONES DE ALMACENAMIENTO.
- v EN LA “ASIGNACION NO CONTIGUA” UN PROGRAMA SE DIVIDE EN VARIOS BLOQUES O “SEGMENTOS” QUE PUEDEN ALMACENARSE EN DIRECCIONES QUE NO TIENEN QUE SER NECESARIAMENTE ADYACENTES:
 - u ES MAS COMPLEJA PERO MAS EFICIENTE QUE LA ASIGNACION CONTINUA.

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO

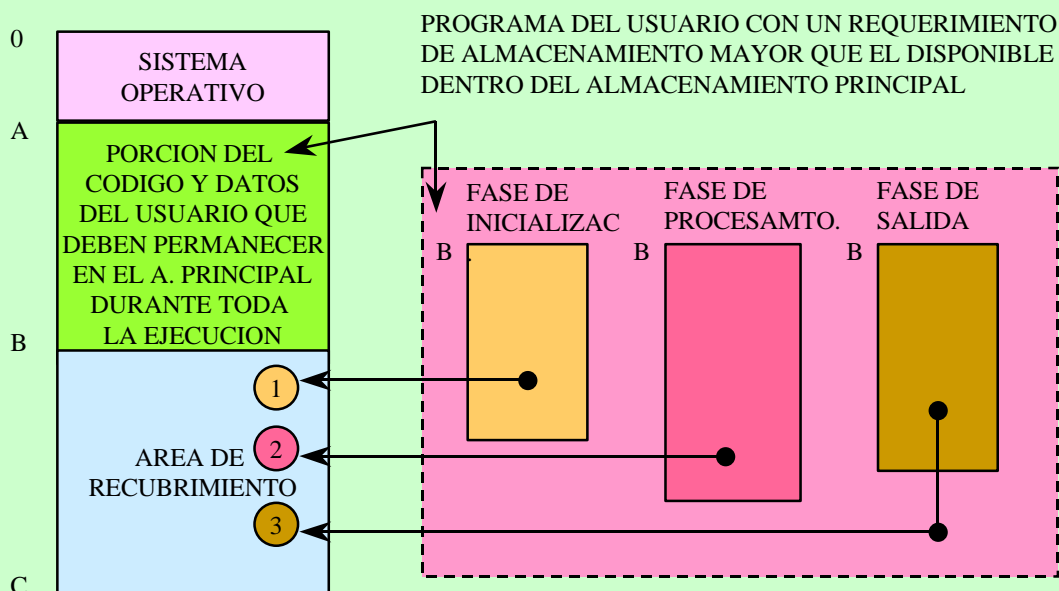
- v **ASIGNACION CONTIGUA DE ALMACENAMIENTO DE UN SOLO USUARIO:**
- v SE CONSIDERAN S. O. QUE YA POSEEN DESARROLLADO EL “SISTEMA DE CONTROL DE ENTRADA / SALIDA”:
 - v IOCS: INPUT / OUTPUT CONTROL SYSTEM.
- v EL TAMAÑO DE LOS PROGRAMAS ESTA LIMITADO POR LA CANTIDAD DE MEMORIA PRINCIPAL:
 - v SE PUEDE SUPERAR ESTE LIMITE CON TECNICAS DE “RECUBRIMIENTOS”:
 - ⊕ SI UNA SECCION PARTICULAR DEL PROGRAMA YA NO ES NECESARIA:
 - SE CARGA OTRA SECCION DESDE EL ALMACENAMIENTO SECUNDARIO OCUPANDO LAS AREAS DE MEMORIA LIBERADAS POR LA SECCION QUE YA NO SE NECESITA.
 - ⊕ LA ADMINISTRACION MANUAL POR PROGRAMA DEL RECUBRIMIENTO ES COMPLICADA Y DIFICULTA EL DESARROLLO Y EL MANTENIMIENTO.
- v PROTECCION EN LOS SISTEMAS DE UN SOLO USUARIO:
- v EL USUARIO TIENE UN COMPLETO CONTROL SOBRE LA TOTALIDAD DEL ALMACENAMIENTO PRINCIPAL:
 - v EL ALMACENAMIENTO SE DIVIDE EN PORCIONES QUE CONTIENEN EL S. O., EL PROGRAMA DEL USUARIO Y UNA PORCION SIN USAR.
 - v EL PROGRAMA DEL USUARIO PODRIA DESTRUIR AREAS DEL S. O. QUE PODRIAN:
 - ⊕ DETENER EL SISTEMA.
 - ⊕ PRODUCIR SALIDAS ERRONEAS.
 - v EL S. O. DEBE ESTAR PROTEGIDO CONTRA EL PROCESO USUARIO:
 - ⊕ LA PROTECCION SE INSTRUMENTA MEDIANTE UN “REGISTRO DE LIMITES” INCORPORADO A LA CPU:
 - CONTIENE LA DIRECCION DE LA INSTRUCCION MAS ALTA UTILIZADA POR EL S. O.
 - SI SE INTENTA INGRESAR AL S. O. LA INSTRUCCION ES INTERCEPTADA Y EL PROCESO FINALIZA.

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO

ASIGNACION CONTIGUA DE ALMACENAMIENTO DE UN SOLO USUARIO



UNA ESTRUCTURA DE RECUBRIMIENTO TIPICA



- 1 CARGA LA FASE DE INICIALIZACION EN B Y LA EJECUTA.
- 2 DESPUES CARGA LA FASE DE PROCESAMIENTO EN B Y LA EJECUTA.
- 3 DESPUES CARGA LA FASE DE SALIDA EN B Y LA EJECUTA.

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO

- v PROCESAMIENTO POR LOTES DE FLUJO UNICO:
- v LOS SISTEMAS DE UN SOLO USUARIO SE DEDICAN A UN TRABAJO DURANTE MAS TIEMPO DEL QUE TOMA SU EJECUCION.
- v LOS TRABAJOS REQUIEREN DE:
 - u “TIEMPO DE INSTALACION”: EL NECESARIO PARA PREPARAR EL ENTORNO OPERATIVO REQUERIDO.
 - u “TIEMPO DE DESCARGA”: EL NECESARIO PARA DESMONTAR EL ENTORNO OPERATIVO QUE FUE REQUERIDO.
- v DURANTE LA INSTALACION Y DESCARGA DE LOS TRABAJOS LA CPU NO ESTA EJECUTANDO DICHOS TRABAJOS REQUERIDOS:
 - u AUTOMATIZAR LA “TRANSICION DE TRABAJO A TRABAJO” REDUCE LA CANTIDAD DE TIEMPO PERDIDO ENTRE TRABAJOS.
 - u SURGIERON LOS SISTEMAS DE “PROCESAMIENTO POR LOTES”.
- v EN EL “PROCESAMIENTO POR LOTES DE FLUJO UNICO” LOS TRABAJOS SE AGRUPAN EN “LOTES” ENCOLANDOSE PARA SU EJECUCION.
- v EL “PROCESADOR DE FLUJOS DE TRABAJOS”:
 - u LEE LAS INSTRUCCIONES DEL “LENGUAJE DE CONTROL DE TRABAJOS”.
 - u FACILITA LA PREPARACION DEL TRABAJO SIGUIENTE.
 - u EMITE INSTRUCCIONES AL OPERADOR DEL SISTEMA.
 - u AUTOMATIZA FUNCIONES ANTERIORMENTE MANUALES.
 - u CUANDO FINALIZA UN TRABAJO:
 - Φ EFECTUA LAS “OPERACIONES DE MANTENIMIENTO” APROPIADAS PARA FACILITAR LA TRANSICION DEL SIGUIENTE TRABAJO.

MULTIPROGRAMACION DE PARTICION FIJA

- v LOS SISTEMAS DE UN SOLO USUARIO DESPERDICIAN GRAN CANTIDAD DE RECURSOS COMPUTACIONALES:
 - v CUANDO OCURRE UNA PETICION DE E / S LA CPU NORMALMENTE NO PUEDE CONTINUAR EL PROCESO HASTA QUE CONCLUYA LA OPERACION DE E / S REQUERIDA.
 - v LOS PERIFERICOS DE E / S FRENAN LA EJECUCION DE LOS PROCESOS:
 - ❖ COMPARATIVAMENTE LA CPU ES VARIOS ORDENES DE MAGNITUD MAS RAPIDA QUE LOS DISPOSITIVOS DE E / S.
- v LOS SISTEMAS DE “MULTIPROGRAMACION” PERMITEN QUE VARIOS PROCESOS USUARIOS COMPITAN AL MISMO TIEMPO POR LOS RECURSOS DEL SISTEMA:
 - v UN TRABAJO EN ESPERA DE E / S CEDERA LA CPU A OTRO TRABAJO QUE ESTE LISTO PARA EFECTUAR COMPUTOS.
 - v EXISTE PARALELISMO ENTRE EL PROCESAMIENTO Y LA E / S.
 - v SE INCREMENTA LA UTILIZACION DE LA CPU Y LA CAPACIDAD GLOBAL DE EJECUCION DEL SISTEMA.
 - v ES NECESARIO QUE VARIOS TRABAJOS RESIDAN A LA VEZ EN LA MEMORIA PRINCIPAL.
- v **MULTIPROGRAMACION DE PARTICION FIJA: TRADUCCION Y CARGA ABSOLUTAS:**
- v LAS “PARTICIONES” DEL ALMACENAMIENTO PRINCIPAL:
 - v SON DE TAMAÑO FIJO.
 - v ALOJAN UN PROCESO CADA UNA.
 - v LA CPU SE CAMBIA RAPIDAMENTE ENTRE LOS PROCESOS CREANDO LA ILUSION DE SIMULTANEIDAD.
- v LOS TRABAJOS SE TRADUCIAN CON ENSAMBLADORES Y COMPILADORES ABSOLUTOS PARA SER EJECUTADOS SOLO DENTRO DE UNA PARTICION ESPECIFICA.
- v EL S. O. RESULTA DE IMPLEMENTACION RELATIVAMENTE SENCILLA

PERO NO SE OPTIMIZA LA UTILIZACION DE LA MEMORIA.

MULTIPROGRAMACION DE PARTICION FIJA

- v **MULTIPROGRAMACION DE PARTICION FIJA: TRADUCCION Y CARGA RELOCALIZABLES:**
- v LOS COMPILADORES, ENSAMBLADORES Y CARGADORES DE RELOCALIZACION:
 - v SE USAN PARA PRODUCIR PROGRAMAS RELOCALIZABLES QUE PUEDAN SER EJECUTADOS EN CUALQUIER PARTICION DISPONIBLE DE TAMAÑO SUFICIENTE PARA ACEPTARLOS.
 - v SON MAS COMPLEJOS QUE LOS ABSOLUTOS.
 - v MEJORAN LA UTILIZACION DEL ALMACENAMIENTO.
 - v CONFIEREN MAS FLEXIBILIDAD EN EL ARMADO DE LA CARGA DE PROCESOS.
- v **PROTECCION EN LOS SISTEMAS DE MULTIPROGRAMACION:**
- v SI SE UTILIZA ASIGNACION CONTIGUA DE MEMORIA LA PROTECCION SUELE IMPLEMENTARSE CON VARIOS “REGISTROS DE LIMITES”.
- v LOS EXTREMOS SUPERIOR E INFERIOR DE UNA PARTICION PUEDEN SER:
 - v DELINEADOS CON DOS REGISTROS.
 - v INDICADOS EL LIMITE INFERIOR O SUPERIOR Y EL TAMAÑO DE LA PARTICION O REGION.
- v **FRAGMENTACION EN LA MULTIPROGRAMACION DE PARTICION FIJA:**
- v LA “FRAGMENTACION DE ALMACENAMIENTO” OCURRE EN TODOS LOS SISTEMAS INDEPENDIENTEMENTE DE SU ORGANIZACION DE MEMORIA.
- v EN LOS S. O. DE MULTIPROGRAMACION DE PARTICION FIJA LA FRAGMENTACION SE PRODUCE CUANDO:
 - v LOS TRABAJOS DEL USUARIO NO LLENAN COMPLETAMENTE SUS PARTICIONES DESIGNADAS.
 - v UNA PARTICION PERMANECE SIN USAR PORQUE ES DEMASIADO PEQUEÑA PARA ALOJAR UN TRABAJO QUE ESTA EN ESPERA.

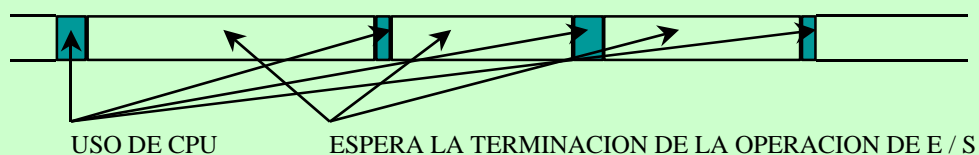
MULTIPROGRAMACION DE PARTICION FIJA

UTILIZACION DE LA CPU EN UN SISTEMA DE UN SOLO USUARIO

PARA UN USUARIO QUE REALIZA CALCULOS INTENSIVOS:

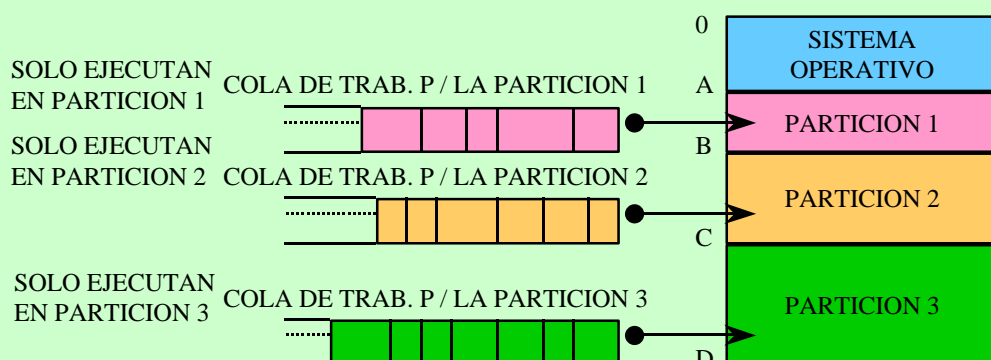


PARA UN USUARIO QUE REALIZA OPERACIONES REGULARES DE E / S:



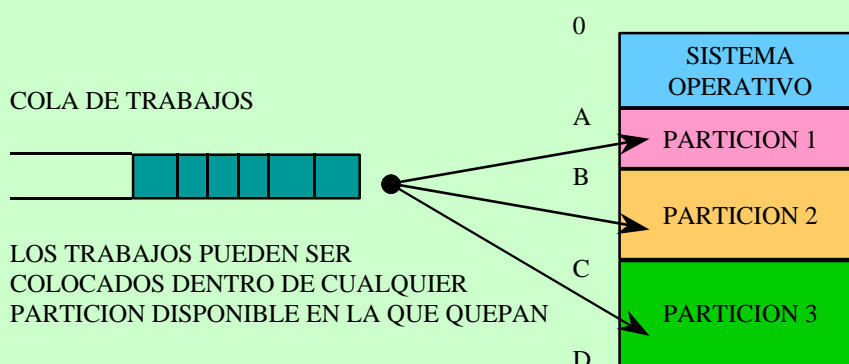
NOTA: FRECUENTEMENTE LA LONGITUD DE LAS ESPERAS DE E / S ES MAS GRANDE EN RELACION CON LA LONGITUD DE LOS PERIODOS DE UTILIZACION DE LA CPU DE LO QUE INDICA ESTE DIAGRAMA. SE DEBE A LA RELACION DE VELOCIDADES ENTRE LA CPU Y LOS DISPOSITIVOS DE E / S.

MULTIPROGRAMACION DE PARTICION FIJA CON TRADUCCION Y CARGA ABSOLUTAS

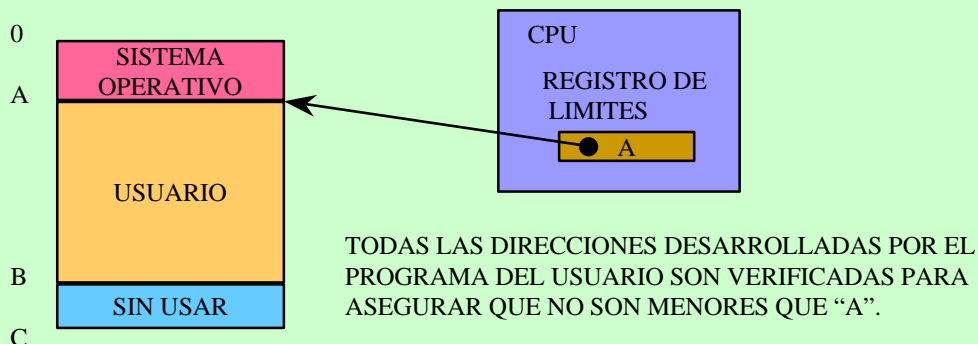


MULTIPROGRAMACION DE PARTICION FIJA

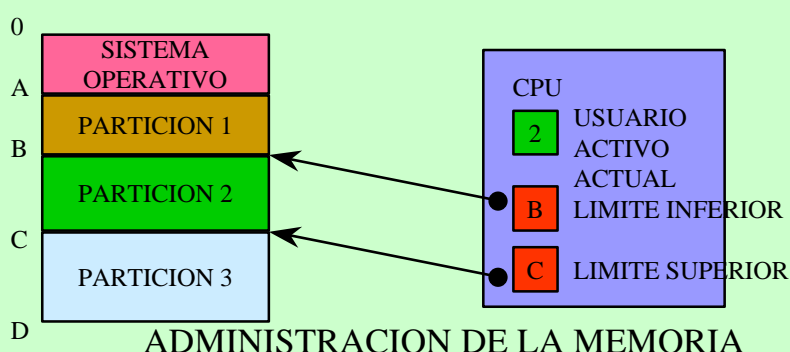
MULTIPROGRAMACION DE PARTICION FIJA CON TRADUCCION Y CARGA RELOCALIZABLES



PROTECCION DEL ALMACENAMIENTO CON ASIGNACION CONTIGUA DE UN SOLO PROCESO DE USUARIO



PROTECCION DEL ALMACENAMIENTO CON ASIGNACION CONTIGUA EN SISTEMAS DE MULTIPROGRAMACION



MULTIPROGRAMACION DE PARTICION VARIABLE

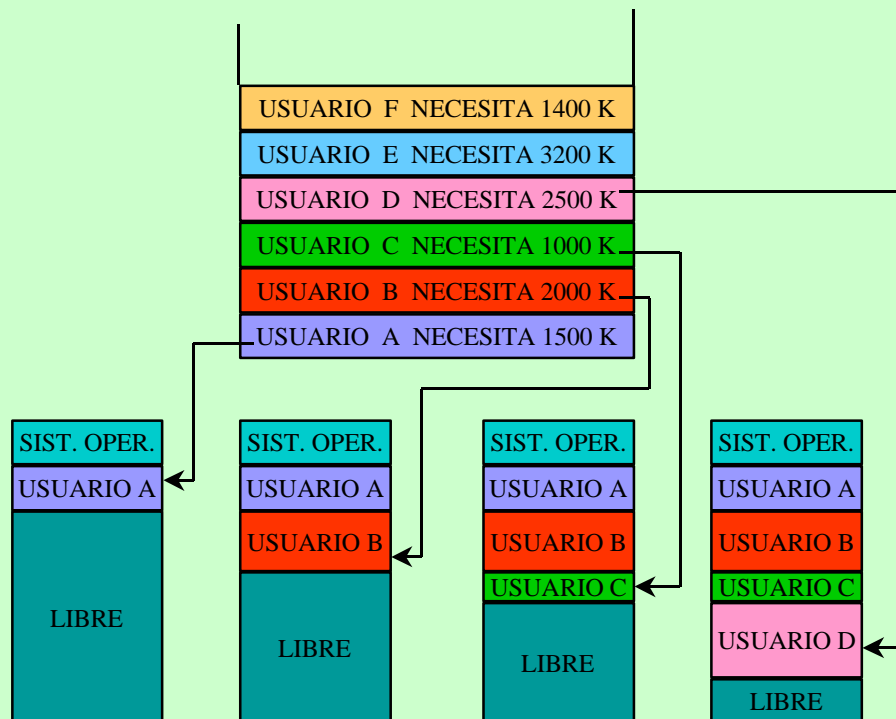
- v LOS PROCESOS OCUPAN TANTO ESPACIO COMO NECESITAN:
 - u NO DEBEN SUPERAR EL ESPACIO DISPONIBLE DE MEMORIA.
- v NO HAY LIMITES FIJOS DE MEMORIA:
 - u LA PARTICION DE UN TRABAJO ES SU PROPIO TAMAÑO.
- v SE CONSIDERAN “ESQUEMAS DE ASIGNACION CONTIGUA”:
 - u UN PROGRAMA DEBE OCUPAR POSICIONES ADYACENTES DE ALMACENAMIENTO.
- v LOS PROCESOS QUE TERMINAN DEJAN DISPONIBLES ESPACIOS DE MEMORIA PRINCIPAL LLAMADOS “AGUJEROS”:
 - u PUEDEN SER USADOS POR OTROS TRABAJOS QUE CUANDO FINALIZAN DEJAN OTROS “AGUJEROS” MENORES:
 - u EN SUCEIVOS PASOS LOS “AGUJEROS” SON CADA VEZ MAS NUMEROSOS PERO MAS PEQUEÑOS:
 - Φ SE GENERA UN DESPERDICIO DE MEMORIA PRINCIPAL.
- v **COMBINACION DE AGUJEROS (AREAS LIBRES):**
- v CONSISTE EN FUSIONAR AGUJEROS ADYACENTES PARA FORMAR UNO SENCILLO MAS GRANDE.
- v SE PUEDE HACER CUANDO UN TRABAJO TERMINA Y EL ALMACENAMIENTO QUE LIBERA TIENE LIMITES CON OTROS AGUJEROS.
- v **COMPRESION O COMPACTACION DE ALMACENAMIENTO:**
- v PUEDE OCURRIR QUE LOS AGUJEROS (AREAS LIBRES) SEPARADOS DISTRIBUIDOS POR TODO EL ALMACENAMIENTO PRINCIPAL CONSTITUYAN UNA CANTIDAD IMPORTANTE DE MEMORIA:
 - u PODRIA SER SUFICIENTE (EL TOTAL GLOBAL DISPONIBLE) PARA ALOJAR A PROCESOS ENCOLADOS EN ESPERA DE MEMORIA.
 - u PODRIA NO SER SUFICIENTE NINGUN AREA LIBRE INDIVIDUAL.
- v LA TECNICA DE COMPRESION DE MEMORIA IMPLICA PASAR TODAS LAS AREAS OCUPADAS DEL ALMACENAMIENTO A UNO DE LOS EXTREMOS DE LA MEMORIA PRINCIPAL:
 - u DEJA UN SOLO AGUJERO GRANDE DE MEMORIA LIBRE CONTIGUA.
 - u ESTA TECNICA SE DENOMINA “RECOGIDA DE RESIDUOS”.

MULTIPROGRAMACION DE PARTICION VARIABLE

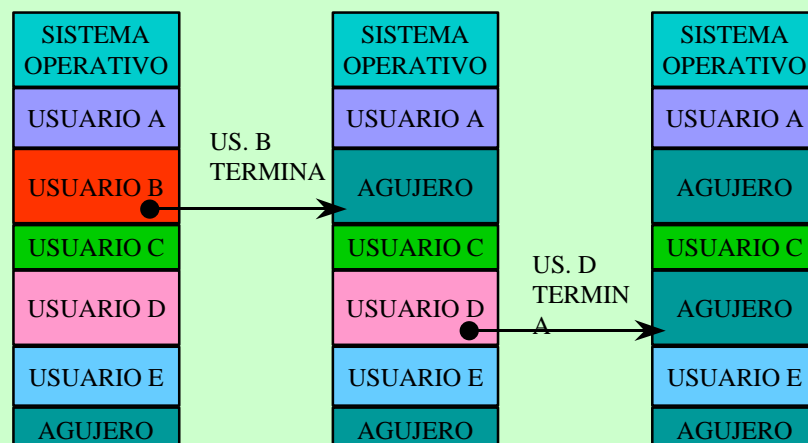
- v PRINCIPALES DESVENTAJAS DE LA COMPRESION:
 - u CONSUME RECURSOS DEL SISTEMA.
 - u EL SISTEMA DEBE DETENER TODO MIENTRAS EFECTUA LA COMPRESION:
 - Φ PUEDE AFECTAR LOS TIEMPOS DE RESPUESTA.
 - u IMPLICA LA RELOCALIZACION (REUBICACION) DE LOS PROCESOS QUE SE ENCUENTRAN EN LA MEMORIA:
 - Φ LA INFORMACION DE RELOCALIZACION DEBE SER DE ACCESIBILIDAD INMEDIATA.
 - u UNA ALTA CARGA DE TRABAJO SIGNIFICA MAYOR FRECUENCIA DE COMPRESION QUE INCREMENTA EL USO DE RECURSOS.
- v **ESTRATEGIAS DE COLOCACION DEL ALMACENAMIENTO:**
- v SE UTILIZAN PARA DETERMINAR EL LUGAR DE LA MEMORIA DONDE SERAN COLOCADOS LOS PROGRAMAS Y DATOS QUE VAN LLEGANDO.
- v “ESTRATEGIA DE MEJOR AJUSTE”:
 - u UN TRABAJO NUEVO ES COLOCADO EN EL AGUJERO EN EL CUAL QUEPA DE FORMA MAS AJUSTADA:
 - Φ DEBE DEJARSE EL MENOR ESPACIO SIN USAR.
- v “ESTRATEGIA DE PRIMER AJUSTE”:
 - u UN TRABAJO NUEVO ES COLOCADO EN EL PRIMER AGUJERO DISPONIBLE CON TAMAÑO SUFICIENTE PARA ALOJARLO.
- v “ESTRATEGIA DE PEOR AJUSTE”:
 - u CONSISTE EN COLOCAR UN PROGRAMA EN EL AGUJERO EN EL QUE QUEPA DE LA PEOR MANERA, ES DECIR EN EL MAS GRANDE POSIBLE:
 - Φ EL AGUJERO RESTANTE ES TAMBIEN GRANDE PARA PODER ALOJAR A UN NUEVO PROGRAMA RELATIVAMENTE GRANDE.

MULTIPROGRAMACION DE PARTICION VARIABLE

ASIGNACION DE PARTICIONES INICIALES EN LA PROGRAMACION DE PARTICION VARIABLE

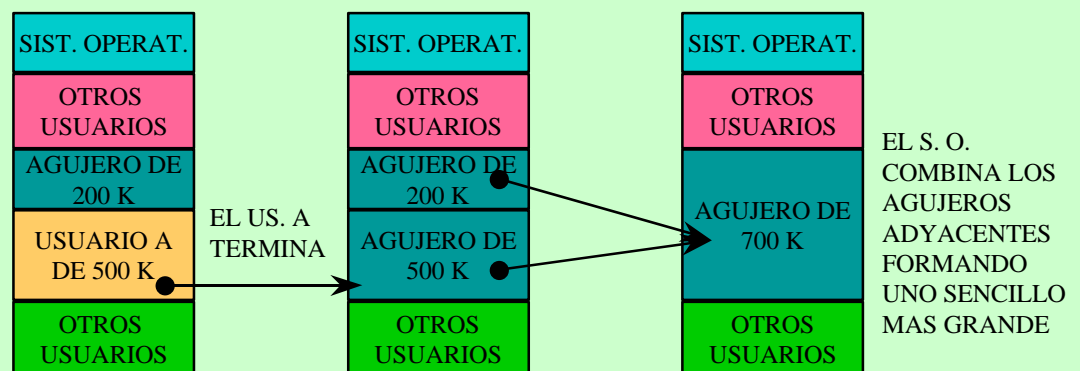


“AGUJEROS” DEL ALMACENAMIENTO EN LA MULTIPROGRAMACION DE PARTICION VARIABLE

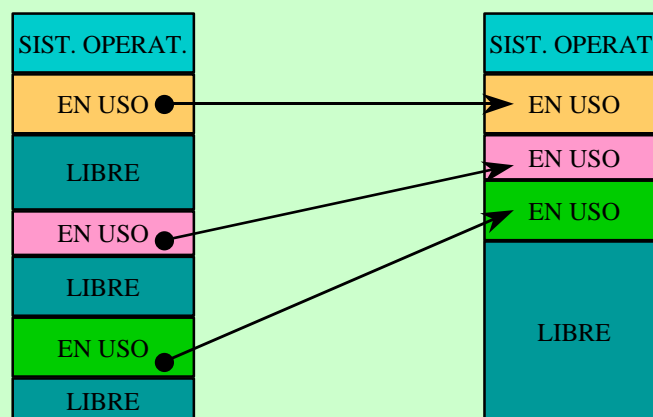


MULTIPROGRAMACION DE PARTICION VARIABLE

COMBINACION DE “AGUJEROS” ADYACENTES DE ALMACENAMIENTO EN LA MULTIPROGRAMACION DE PARTICION VARIABLE



COMPRESION (COMPACTACION) DE ALMACENAMIENTO EN LA MULTIPROGRAMACION DE PARTICION VARIABLE



EL S. O. COLOCA TODOS LOS BLOQUES “EN USO” JUNTOS, DEJANDO EL ALMACENAMIENTO LIBRE COMO UN UNICO AGUJERO GRANDE.

MULTIPROGRAMACION CON INTERCAMBIO DE ALMACENAMIENTO

- v EN EL ESQUEMA DE “INTERCAMBIO” LOS PROGRAMAS DEL USUARIO NO REQUIEREN PERMANECER EN LA MEMORIA PRINCIPAL HASTA SU TERMINACION.
- v UNA VARIANTE CONSISTE EN QUE UN TRABAJO SE EJECUTA HASTA QUE YA NO PUEDE CONTINUAR:
 - v CEDE EL ALMACENAMIENTO Y LA CPU AL SIGUIENTE TRABAJO.
 - v LA TOTALIDAD DEL ALMACENAMIENTO SE DEDICA A UN TRABAJO DURANTE UN BREVE PERIODO DE TIEMPO.
 - v LOS TRABAJOS SON “INTERCAMBIADOS”:
 - Φ UN TRABAJO PUEDE SER INTERCAMBIADO VARIAS VECES ANTES DE LLEGAR A SU TERMINACION.
- v ES UN ESQUEMA RAZONABLE Y EFICIENTE PARA UN NUMERO RELATIVAMENTE REDUCIDO DE PROCESOS DE USUARIOS.
- v LOS SISTEMAS DE INTERCAMBIO FUERON LOS PREDECESORES DE LOS SISTEMAS DE PAGINACION.
- v EL RENDIMIENTO DE LOS SISTEMAS DE INTERCAMBIO MEJORA AL REDUCIR EL TIEMPO DE INTERCAMBIO:
 - v MANTENIENDO AL MISMO TIEMPO VARIAS “IMAGENES DE USUARIO” EN LA MEMORIA PRINCIPAL.
 - v RETIRANDO UNA IMAGEN DE USUARIO DE LA MEMORIA PRINCIPAL SOLO CUANDO ES NECESARIO SU ALMACENAMIENTO PARA UNA NUEVA IMAGEN.
 - v INCREMENTANDO LA CANTIDAD DE MEMORIA PRINCIPAL DISPONIBLE EN EL SISTEMA.
- v LAS IMAGENES DE USUARIO (IMAGENES DE MEMORIA) RETIRADAS DEL ALMACENAMIENTO PRINCIPAL SE GRABAN EN EL ALMACENAMIENTO SECUNDARIO (DISCOS).

ORGANIZACION DEL ALMACENAMIENTO VIRTUAL:

INTRODUCCION

- v “ALMACENAMIENTO VIRTUAL” SIGNIFICA LA CAPACIDAD DE DIRECCIONAR UN ESPACIO DE ALMACENAMIENTO MUCHO MAYOR QUE EL DISPONIBLE EN EL ALMACENAMIENTO PRIMARIO DE DETERMINADO SISTEMA DE COMPUTACION.
- v ESTA TECNOLOGIA APARECIO EN 1960 EN LA UNIVERSIDAD DE MANCHESTER (INGLATERRA), EN EL SISTEMA “ATLAS”.
- v LOS METODOS MAS COMUNES DE IMPLEMENTACION SON MEDIANTE:
 - υ TECNICAS DE “PAGINACION”.
 - υ TECNICAS DE “SEGMENTACION”.
 - υ UNA COMBINACION DE AMBAS TECNICAS.
- v LAS DIRECCIONES GENERADAS POR LOS PROGRAMAS EN SU EJECUCION NO SON, NECESARIAMENTE, AQUELLAS CONTENIDAS EN EL ALMACENAMIENTO PRIMARIO (MEMORIA REAL):
 - υ LAS DIRECCIONES VIRTUALES SUELEN SELECCIONARSE DENTRO DE UN N° MUCHO MAYOR DE DIRECCIONES QUE LAS DISPONIBLES DENTRO DEL ALMACENAMIENTO PRIMARIO.
- v **EVOLUCION EN LAS ORGANIZACIONES DE ALMACENAMIENTO:**
- v **REAL:**
 - υ SISTEMAS DEDICADOS A UN SOLO USUARIO.
- v **REAL:**
 - υ SISTEMAS DE MULTIPROGRAMACION EN MEMORIA REAL:
 - Φ MULTIPROGRAMACION EN PARTICION FIJA:
 - ABSOLUTA.
 - RELOCALIZABLE (REUBICABLE).
 - Φ MULTIPROGRAMACION EN PARTICION VARIABLE.
- v **VIRTUAL:**
 - υ MULTIPROGRAMACION EN ALMACENAMIENTO VIRTUAL:
 - Φ PAGINACION PURA.
 - Φ SEGMENTACION PURA.
 - Φ COMBINACION PAGINACION / SEGMENTACION.

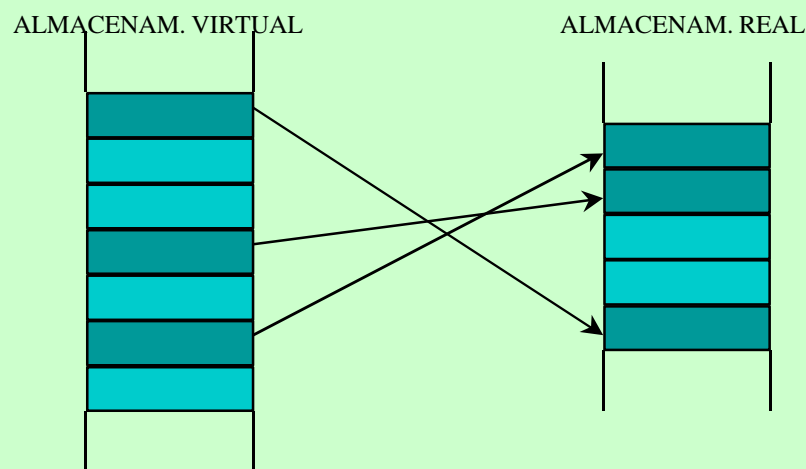
CONCEPTOS BASICOS DE ALMACENAMIENTO VIRTUAL

- v LA CLAVE DEL CONCEPTO DE MEMORIA (ALMACENAMIENTO) VIRTUAL ESTA EN LA DISOCIACION:
 - u DE LAS DIRECCIONES A LAS QUE HACE REFERENCIA UN PROGRAMA.
 - u DE LAS DIRECCIONES DISPONIBLES EN LA MEMORIA REAL (ALMACENAMIENTO PRIMARIO).
- v “DIRECCIONES VIRTUALES”:
 - u SON LAS REFERIDAS POR UN PROCESO EN EJECUCION.
- v “DIRECCIONES REALES”:
 - u SON LAS DISPONIBLES DENTRO DEL ALMACENAMIENTO PRIMARIO.
- v “ESPACIO DE DIRECCIONES VIRTUALES (V)” DE UN PROCESO:
 - u ES EL N° DE DIRECCIONES VIRTUALES A QUE PUEDE HACER REFERENCIA EL PROCESO.
- v “ESPACIO DE DIRECCIONES REALES (R)” DE UN COMPUTADOR:
 - u ES EL N° DE DIRECCIONES REALES DISPONIBLES EN EL ORDENADOR.
- v LOS PROCESOS HACEN REFERENCIA A DIRECCIONES VIRTUALES PERO ESTAS DEBEN EJECUTARSE EN EL ALMACENAMIENTO REAL:
 - u LAS DIRECCIONES VIRTUALES DEBEN SER TRANSFORMADAS DENTRO DE LAS DIRECCIONES REALES, MIENTRAS EL PROCESO ESTA EN EJECUCION:
 - ❖ LA TRADUCCION DE DIRECCIONES DEBERA HACERSE RAPIDAMENTE PARA NO DEGRADAR AL SISTEMA.
- v EXISTEN VARIOS MEDIOS PARA ASOCIAR LAS DIRECCIONES VIRTUALES CON LAS REALES.
- v LOS MECANISMOS DE “TRADUCCION DINAMICA DE DIRECCIONES” (DAT) CONVIERTEN LAS DIRECCIONES VIRTUALES EN REALES AL EJECUTARSE EL PROCESO.
- v LAS DIRECCIONES CONTIGUAS DENTRO DEL ESPACIO DE DIRECCIONES VIRTUALES DE UN PROCESO NO TIENEN POR QUE SER CONTIGUAS

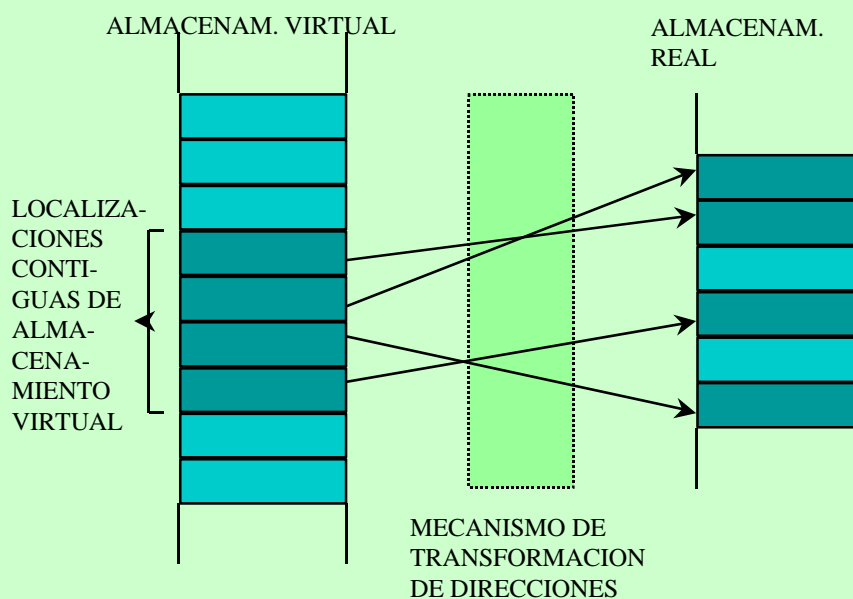
DENTRO DEL ALMACENAMIENTO REAL: “CONTIGÜIDAD ARTIFICIAL”.

CONCEPTOS BASICOS DE ALMACENAMIENTO VIRTUAL

TRANSFORMACION DE ITEMS DEL ESPACIO DE DIRECCIONES VIRTUALES AL ESPACIO DE DIRECCIONES REALES



CONTIGÜIDAD ARTIFICIAL

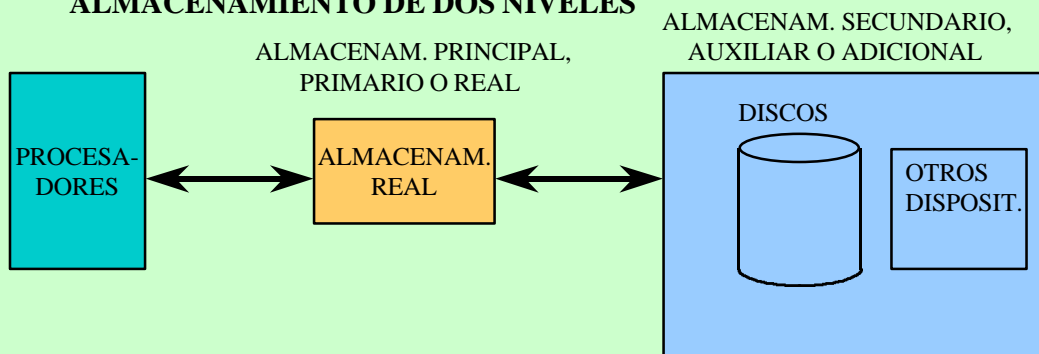


ORGANIZACION DEL ALMACENAMIENTO DE NIVELES MULTIPLES

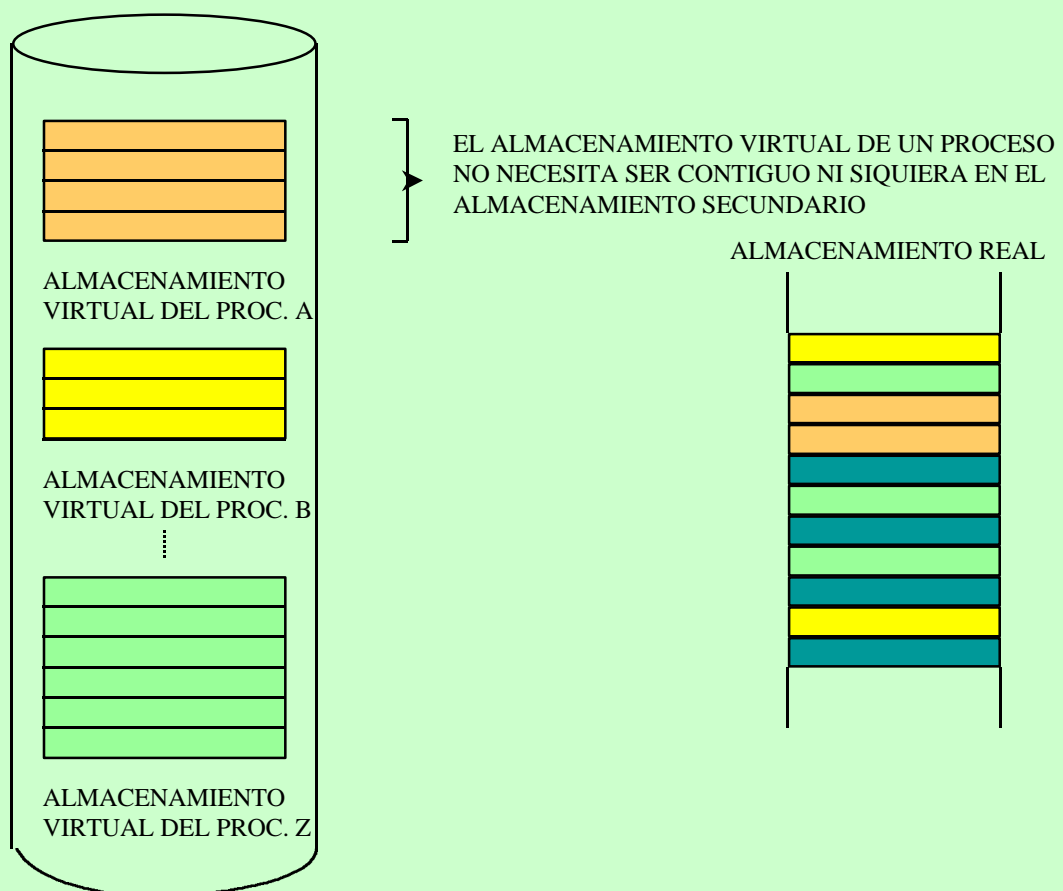
- v SE DEBEN PROPORCIONAR LOS MEDIOS PARA RETENER PROGRAMAS Y DATOS EN UN GRAN ALMACENAMIENTO AUXILIAR PARA:
 - v PERMITIR QUE EL ESPACIO DE DIRECCIONES VIRTUALES DE UN USUARIO SEA MAYOR QUE EL ESPACIO DE DIRECCIONES REALES.
 - v SOPORTAR MULTIPROGRAMACION DE FORMA EFECTIVA EN UN SISTEMA CON MUCHOS USUARIOS QUE COMPARTAN EL ALMACENAMIENTO REAL.
- v SE UTILIZA UN ESQUEMA DE ALMACENAMIENTO DE DOS NIVELES:
 - v PRIMER NIVEL: “ALMACENAMIENTO REAL”:
 - Φ EN EL SE EJECUTAN LOS PROCESOS Y EN EL DEBEN ESTAR LOS DATOS PARA QUE UN PROCESO PUEDA REFERIRSE A ELLOS.
 - v SEGUNDO NIVEL: “ALMACENAMIENTO AUXILIAR, SECUNDARIO O ADICIONAL”:
 - Φ GENERALMENTE CONSTA DE DISCOS DE GRAN CAPACIDAD QUE PUEDEN MANTENER LOS PROGRAMAS Y DATOS QUE NO CABEN AL MISMO TIEMPO EN EL MAS LIMITADO ALMACENAMIENTO REAL.
- v CUANDO SE VA A EJECUTAR UN PROCESO SU CODIGO Y DATOS SE PASAN AL ALMACENAMIENTO PRINCIPAL.
- v EL ALMACENAMIENTO REAL ES COMPARTIDO POR VARIOS PROCESOS:
 - v CADA PROCESO PUEDE TENER UN ESPACIO DE DIRECCIONES VIRTUALES MUCHO MAYOR QUE EL ALMACENAMIENTO REAL:
 - Φ SOLO SE MANTIENE AL MISMO TIEMPO UNA PEQUEÑA PARTE DE LOS PROGRAMAS Y DATOS DE CADA PROCESO EN EL ALMACENAMIENTO REAL.

ORGANIZACION DEL ALMACENAMIENTO DE NIVELES MULTIPLES

ALMACENAMIENTO DE DOS NIVELES



ALMACENAMIENTO AUXILIAR



TRANSFORMACION DE BLOQUES

- v LOS MECANISMOS DE TRADUCCION DINAMICA DE DIRECCIONES DEBEN MANTENER “MAPAS” QUE ILUSTREN:
 - u QUE DIRECCIONES DEL ALMACENAMIENTO VIRTUAL SE ENCUENTRAN EN EL ALMACENAMIENTO REAL.
 - u DONDE SE ENCUENTRAN.
- v LA INFORMACION SE AGRUPA EN “BLOQUES”:
 - u EL SISTEMA ESTA INFORMADO DEL LUGAR DEL ALMACENAMIENTO REAL DONDE HAN SIDO COLOCADOS LOS BLOQUES DE ALMACENAMIENTO VIRTUAL.
 - u CUANTO MAYOR SEA EL BLOQUE MENOR SERA LA FRACCION DEL ALMACENAMIENTO REAL QUE DEBE DEDICARSE A CONTENER LA INFORMACION DEL MAPA.
 - u CON BLOQUES GRANDES:
 - Φ SE REDUCE LA SOBRECARGA DE ALMACENAMIENTO DEL MECANISMO DE TRANSFORMACION.
 - Φ SE INCREMENTA EL TIEMPO DE TRANSFERENCIA ENTRE LOS ALMACENAMIENTOS SECUNDARIO Y PRIMARIO.
 - Φ CONSUMEN MAS ALMACENAMIENTO REAL PUDIENDO LIMITAR EL N° DE PROCESOS QUE PUEDEN COMPARTIRLO.
 - u LOS BLOQUES PUEDEN SER DE TAMAÑO:
 - Φ IGUAL: SE DENOMINAN “PAGINAS” Y LA ORGANIZACION DE ALMACENAMIENTO VIRTUAL ASOCIADA SE DENOMINA “PAGINACION”.
 - Φ DIFERENTE: SE DENOMINAN “SEGMENTOS” Y LA ORGANIZACION DE ALMACENAMIENTO VIRTUAL ASOCIADA SE DENOMINA “SEGMENTACION”.
 - u SE PUEDEN COMBINAR AMBAS TECNICAS: SEGMENTOS DE TAMAÑO VARIABLE COMPUESTOS DE PAGINAS DE TAMAÑO FIJO.
- v LAS DIRECCIONES SON “BIDIMENSIONALES”:
 - u UNA **DIRECCION VIRTUAL** “V” SE INDICA POR UN PAR ORDENADO “(B,D)”:
 - Φ “B”: N° DEL BLOQUE DONDE RESIDE.
 - Φ “D”: DESPLAZAMIENTO A PARTIR DEL INICIO DEL BLOQUE.

TRANSFORMACION DE BLOQUES

- v LA TRADUCCION DE UNA DIRECCION VIRTUAL $V = (B,D)$ A LA DIRECCION REAL "R" CONSIDERA LO SIGUIENTE:
 - v CADA PROCESO TIENE SU "TABLA DE MAPA DE BLOQUES" MANTENIDA POR EL SISTEMA EN EL ALMACENAMIENTO REAL.
 - v UN REGISTRO ESPECIAL DEL PROCESADOR LLAMADO "REGISTRO ORIGEN DE LA TABLA DE BLOQUES" SE CARGA CON LA DIRECCION REAL "A" DE LA "TABLA DE MAPA DE BLOQUES":
 - ⊕ CONTIENE UNA ESTRADA PARA CADA BLOQUE DEL PROCESO.
 - ⊕ LAS ENTRADAS SE MANTIENEN EN ORDEN SECUENCIAL PARA EL BLOQUE 0, BLOQUE 1, ETC.
 - ⊕ SE AÑADE EL BLOQUE N° "B" A LA DIRECCION BASE "A" DE LA "TABLA DE BLOQUES" PARA FORMAR LA DIRECCION REAL DE LA ENTRADA DE LA "TABLA DE MAPA DE BLOQUES" PARA EL BLOQUE "B":
 - CONTIENE LA DIRECCION REAL "B" ' " PARA EL BLOQUE "B".
 - EL DESPLAZAMIENTO "D" SE AÑADE A LA DIRECCION DE INICIO DEL BLOQUE, "B" ' " PARA FORMAR LA "DIRECCION REAL" DESEADA: $R = B' + D$.
- v LA TRANSFORMACION DE BLOQUES SE EFECTUA EN FORMA DINAMICA MIENTRAS SE EJECUTA UN PROCESO:
 - v SI LA IMPLEMENTACION NO ES EFICIENTE SU SOBRECARGA PUEDE CAUSAR UNA DEGRADACION DEL RENDIMIENTO QUE PODRIA ELIMINAR EN PARTE LAS VENTAJAS DE LA UTILIZACION DEL ALMACENAMIENTO VIRTUAL.

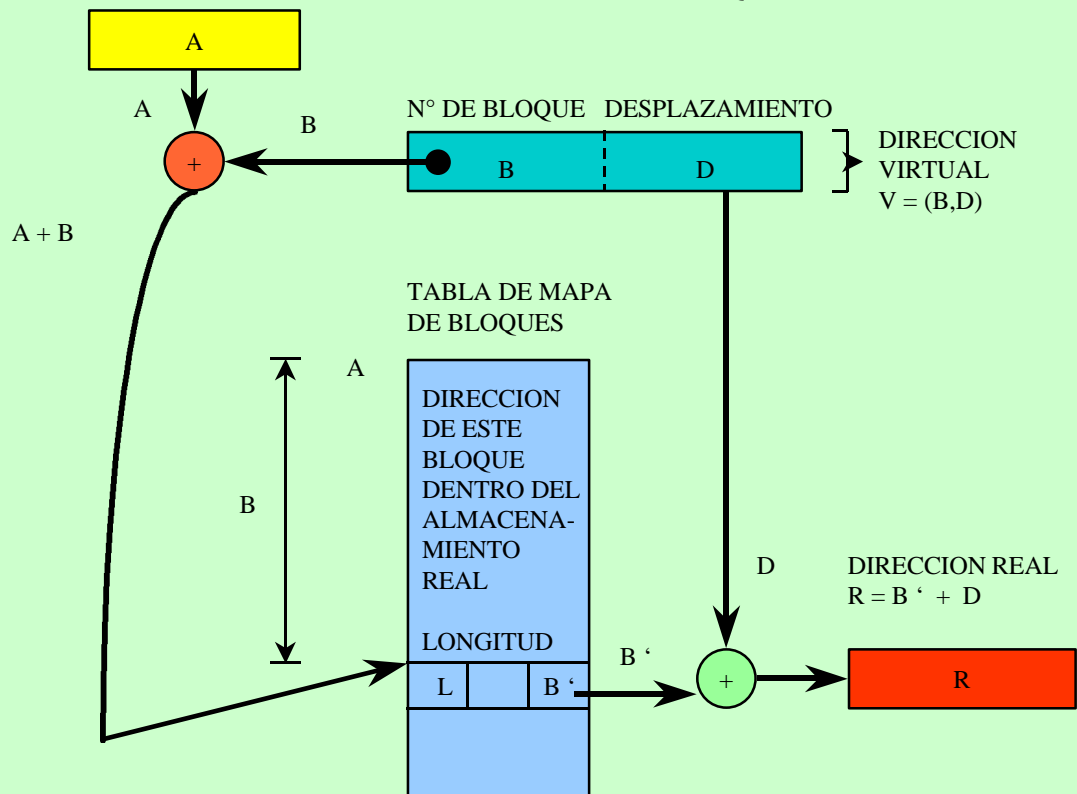
TRANSFORMACION DE BLOQUES

FORMATO DE LA DIRECCION VIRTUAL DENTRO DE UN SISTEMA DE TRANSFORMACION DE BLOQUES



TRADUCCION DE DIRECCIONES VIRTUALES CON TRANSFORMACION DE BLOQUES

REGISTRO DE ORIGEN DE LA TABLA DE BLOQUES, QUE CONTIENE LA DIRECCION BASE DE LA TABLA DE MAPA DE BLOQUES



CONCEPTOS BASICOS DE PAGINACION

- v FRECUENTEMENTE SE DIFERENCIA ENTRE LA “PAGINACION PURA” Y LA “COMBINACION DE PAGINACION Y SEGMENTACION”.
- v LAS PAGINAS SE TRANSFIEREN DEL ALMACENAMIENTO SECUNDARIO AL PRIMARIO EN BLOQUES LLAMADOS “MARCOS DE PAGINAS”:
 - u TIENEN EL MISMO TAMAÑO QUE LAS PAGINAS.
 - u COMIENZAN EN DIRECCIONES DEL ALMACENAMIENTO REAL QUE SON MULTIPLOS ENTEROS DEL TAMAÑO FIJO DE LA PAGINA.
 - u PODRA COLOCARSE UNA NUEVA PAGINA DENTRO DE CUALQUIER “MARCO DE PAGINA” O “CELDA DE PAGINA” DISPONIBLE.
- v LA “TRADUCCION DINAMICA DE DIRECCIONES” INCLUYE:
 - u UN PROCESO EN EJECUCION HACE REFERENCIA A UNA DIRECCION VIRTUAL “ $V = (P,D)$ ”.
 - u UN MECANISMO DE TRANSFORMACION DE PAGINAS BUSCA LA PAGINA “P” EN LA “TABLA DE PAGINAS”:
 - Φ DETERMINA SI LA PAGINA “P” SE ENCUENTRA EN EL MARCO DE PAGINA “P ‘ “.
 - u LA DIRECCION DE ALMACENAMIENTO REAL SE FORMA POR LA CONCATENACION DE “P ‘ “ Y “D”.
- v LA TABLA DE “MAPA DE PAGINAS” DEBE INDICAR SI SE ENCUENTRA O NO EN EL ALMACENAMIENTO PRIMARIO LA PAGINA REFERENCIADA:
 - u EN CASO AFIRMATIVO DONDE ESTA EN LA MEMORIA REAL.
 - u EN CASO NEGATIVO DONDE PUEDE ESTAR EN EL ALMACENAMIENTO SECUNDARIO.
- v LA DIRECCION DE ALMACENAMIENTO PRIMARIO “A”, DONDE COMIENZA EL MARCO DE PAGINA “P ‘ “ (SUPONIENDO UN TAMAÑO DE PAGINA “P”), ESTA DADA POR: “ $A = (P) (P ‘)$ ”:
 - u SE SUPONE MARCOS DE PAGINA NUMERADOS 0, 1, 2, ETC.

CONCEPTOS BASICOS DE PAGINACION

FORMATO DE LA DIRECCION VIRTUAL EN UN SISTEMA DE PAGINACION PURA

PAGINA N° "P"	DESPLAZAMIENTO "D"	DIRECCION VIRTUAL $V = (P,D)$
---------------	--------------------	-------------------------------------

ALMACENAMIENTO REAL DIVIDIDO EN MARCOS DE PAGINAS

0	MARCO DE PAG. 0			
P	MARCO DE PAG. 1	<u>Nº DE</u>	<u>TAMAÑO DE</u>	<u>Nº DE DIRECCIONES DE</u>
2P	MARCO DE PAG. 2	<u>PAGINA</u>	<u>PAGINA</u>	<u>ALMACENAMIENTO REAL</u>
3P	MARCO DE PAG. 3	0	P	0 --> P - 1
		1	P	P --> 2P - 1
4P	MARCO DE PAG. 4	2	P	2P --> 3P - 1
		3	P	3P --> 4P - 1
5P	MARCO DE PAG. 5	4	P	4P --> 5P - 1
		5	P	5P --> 6P - 1
6P	MARCO DE PAG. 6	6	P	6P --> 7P - 1
		7	P	7P --> 8P - 1
7P	MARCO DE PAG. 7	.		
		.		
		.		

UNA ENTRADA DE LA TABLA DE MAPA DE PAGINAS

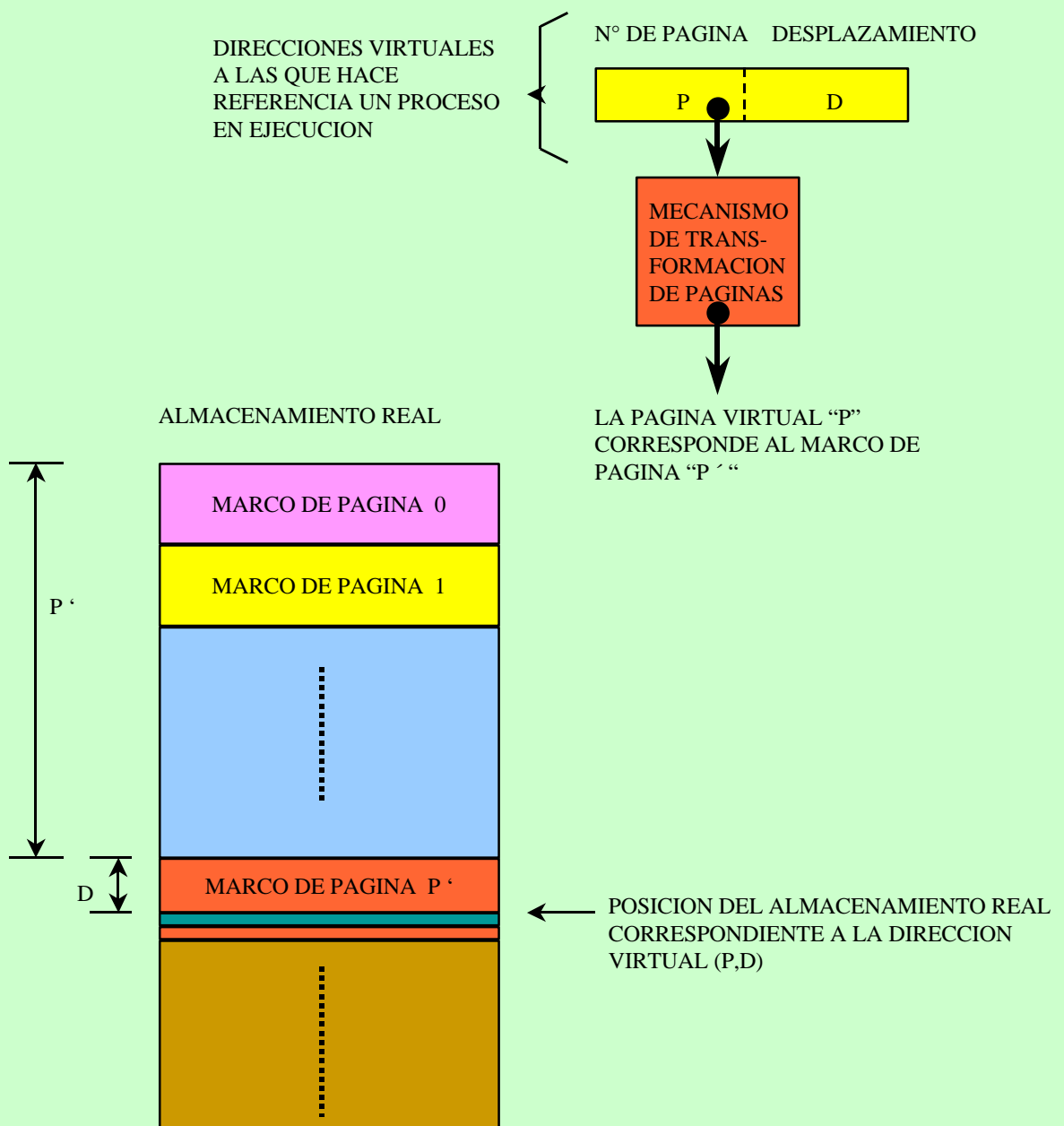
BIT DE RESIDENCIA DE PAGINA	DIRECCION DE ALMACENA- MIENTO SECUNDARIO (SI LA PAGINA NO ESTA EN EL ALMACENAMIENTO REAL)	Nº DEL MARCO DE PAGINA (SI LA PAGINA ESTA EN EL ALMACENAMIENTO REAL)
-----------------------------------	--	--

R	S	P'
---	---	----

R = 0 SI LA PAGINA NO ESTA EN EL ALMACENAMIENTO REAL
R = 1 SI LA PAGINA ESTA EN EL ALMACENAMIENTO REAL

CONCEPTOS BASICOS DE PAGINACION

CORRESPONDENCIA ENTRE LAS DIRECCIONES DE ALMACENAMIENTO VIRTUAL Y LAS DIRECCIONES DE ALMACENAMIENTO REAL EN UN SISTEMA DE PAGINACION

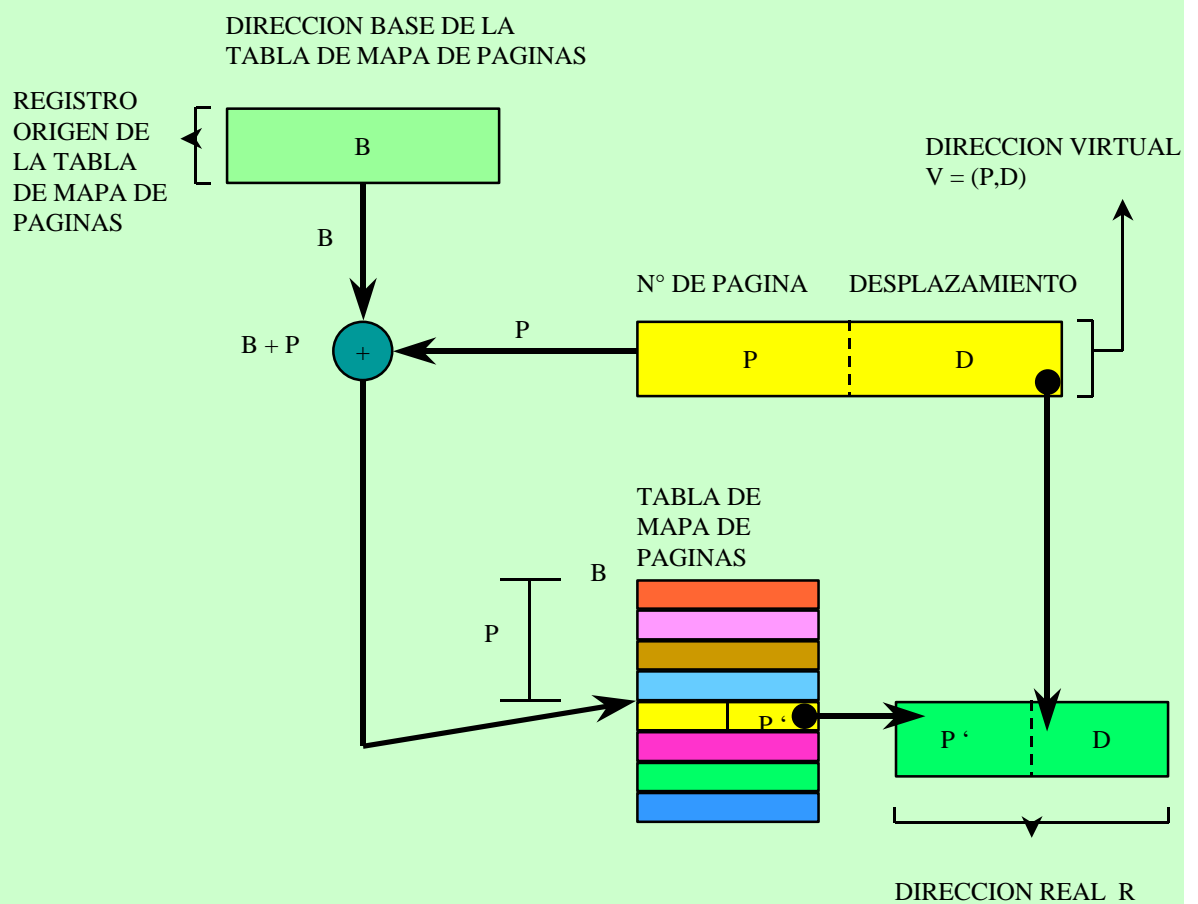


CONCEPTOS BASICOS DE PAGINACION

- v **TRADUCCION DE DIRECCIONES DE PAGINACION POR TRANSFORMACION DIRECTA:**
- v UN PROCESO EN EJECUCION HACE REFERENCIA A LA DIRECCION VIRTUAL $V = (P, D)$.
- v ANTES QUE UN PROCESO COMIENZE SU EJECUCION, EL S. O. CARGA LA DIRECCION DE ALMACENAMIENTO PRIMARIO DE LA "TABLA DE MAPA DE PAGINAS" EN EL "REGISTRO ORIGEN DE LA TABLA DE MAPA DE PAGINAS".
- v LA DIRECCION BASE DE LA TABLA DE MAPA DE PAGINAS ES "B".
- v EL N° DE PAGINA ES "P".
- v LA DIRECCION EN EL ALMACENAMIENTO PRIMARIO DE LA ENTRADA EN LA TABLA DE MAPA DE PAGINAS PARA LA PAGINA "P" ES "B + P":
 - u INDICA QUE EL MARCO DE PAGINA "P" CORRESPONDE A LA PAGINA VIRTUAL.
 - u "P" SE CONCATENA CON EL DESPLAZAMIENTO "D" PARA FORMAR LA DIRECCION REAL "R".
- v "ESTO ES UN EJ. DE TRANSFORMACION DIRECTA DEBIDO A QUE LA TABLA DE MAPA DE PAGINAS CONTIENE UNA ENTRADA POR CADA UNA DE LAS PAGINAS DEL ALMACENAMIENTO VIRTUAL DE ESTE PROCESO".
- v LA DIRECCION VIRTUAL QUE SE ESTA TRADUCIENDO Y LA DIRECCION BASE DE LA TABLA DE MAPA DE PAGINAS SON MANTENIDAS EN UN REGISTRO DE ALTA VELOCIDAD DEL CONTROL DEL PROCESADOR.
- v LA TABLA DE MAPA DE PAGINAS TRANSFORMADA DIRECTAMENTE SUELE MANTENERSE EN EL ALMACENAMIENTO PRIMARIO:
 - u LAS REFERENCIAS A ESTA TABLA REQUIEREN UN CICLO COMPLETO DE ALMACENAMIENTO PRIMARIO:
 - Φ GENERALMENTE ES LA PARTE MAS LARGA DE UN CICLO DE EJECUCION DE INSTRUCCIONES.
 - u SE REQUIERE OTRO CICLO DE EJECUCION DE ALMACENAMIENTO PRIMARIO PARA LA TRANSFORMACION DE PAGINAS:
 - Φ PUEDE OCASIONAR DEGRADACION EQUIVALENTE A UN 50%:
 - UNA SOLUCION SERIA TENER LA TABLA COMPLETA DE MAPA DE PAGINAS DE TRANSFORMACION DIRECTA EN LA "CACHE" DE MUY ALTA VELOCIDAD.

CONCEPTOS BASICOS DE PAGINACION

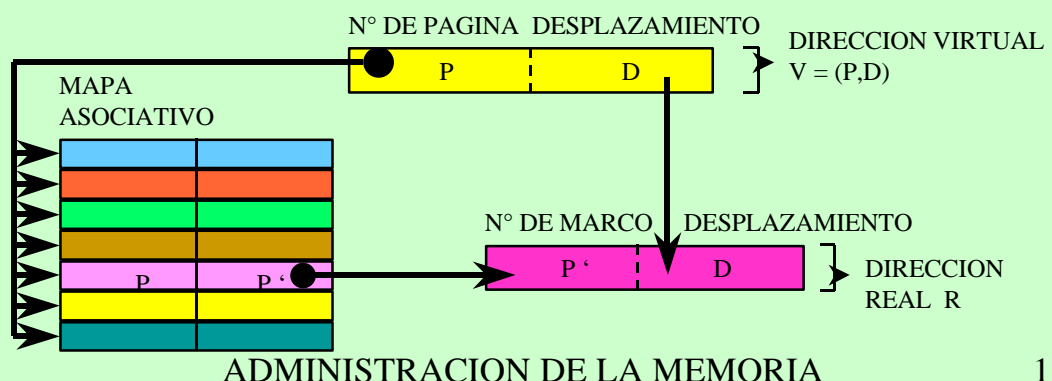
TRADUCCION DE DIRECCIONES DE PAGINAS POR TRANSFORMACION DIRECTA



CONCEPTOS BASICOS DE PAGINACION

- v **TRADUCCION DE DIRECCIONES DE PAGINACION POR TRANSFORMACION ASOCIATIVA:**
- v UNA FORMA DE ACELERAR LA TRADUCCION DINAMICA DE PAGINAS CONSISTE EN COLOCAR LA TABLA COMPLETA DE MAPA DE PAGINAS EN UN “ALMACENAMIENTO ASOCIATIVO” QUE TENGA UN TIEMPO DE CICLO MUCHO MAS RAPIDO QUE EL ALMACENAMIENTO PRIMARIO.
- v UNA VARIANTE ES LA “TRANSFORMACION ASOCIATIVA PURA”.
- v UN PROGRAMA EN EJECUCION HACE REFERENCIA A LA DIRECCION VIRTUAL $V = (P,D)$.
- v CADA ENTRADA EN EL ALMACENAMIENTO ASOCIATIVO SE BUSCA DE FORMA SIMULTANEA PARA LA PAGINA “P”:
- v SE OBTIENE “P’ “ COMO EL MARCO DE PAGINA CORRESPONDIENTE A LA PAGINA “P”.
 - v SE CONCATENA “P’ “ CON “D” FORMANDO LA DIRECCION REAL “R”.
- v CADA UNA DE LAS CELULAS DEL ALMACENAMIENTO ASOCIATIVO SE REGISTRA DE MANERA SIMULTANEA:
- v HACE COSTOSO EL ALMACENAMIENTO ASOCIATIVO.
 - v IMPLEMENTAR LA TRANSFORMACION ASOCIATIVA PURA RESULTA DEMASIADO COSTOSO:
 - ❖ TAL LO OCURRIDO CON LA IMPLEMENTACION DE LA TRANSFORMACION DIRECTA PURA UTILIZANDO “CACHE”.

TRADUCCION DE DIRECCIONES DE PAGINAS POR PLANIFICACION ASOCIATIVA PURA



CONCEPTOS BASICOS DE PAGINACION

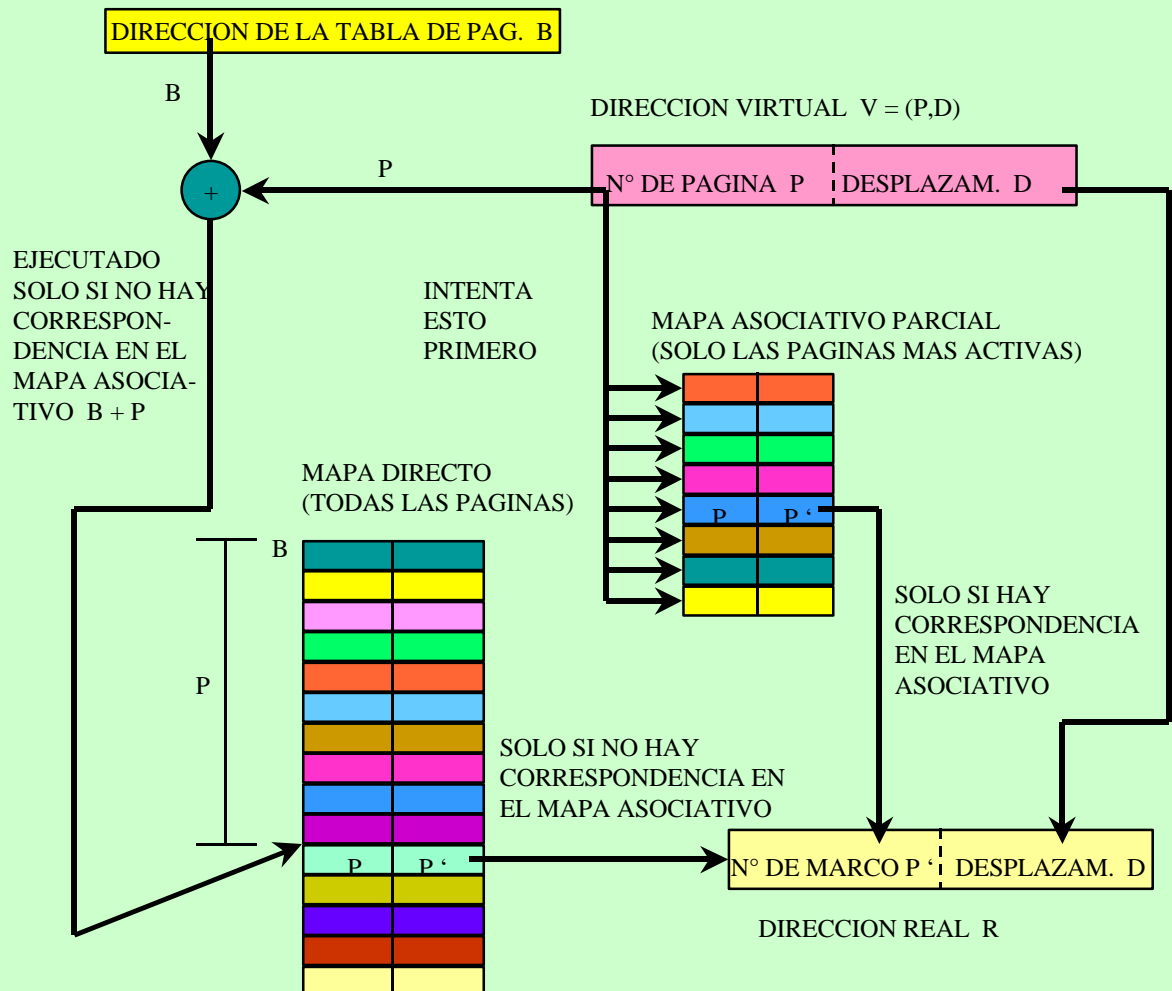
- v **TRADUCCION DE DIRECCIONES DE PAGINACION POR COMBINACION DE TRANSFORMACION ASOCIATIVA / DIRECTA:**
- v SE UTILIZA UN ALMACENAMIENTO ASOCIATIVO CAPAZ DE MANTENER SOLO UN PEQUEÑO PORCENTAJE DEL MAPA COMPLETO DE PAGINA PARA UN PROCESO.
- v LAS ENTRADAS DE PAGINA CONTENIDAS EN ESTE MAPA REDUCIDO CORRESPONDEN SOLO A LAS PAGINAS REFERENCIADAS RECIENTEMENTE:
 - u SE PRESUPONE QUE UNA PAGINA RECIENTEMENTE REFERENCIADA TENDRA POSIBILIDADES DE SERLO DE NUEVO PROXIMAMENTE.
 - u LOS RENDIMIENTOS OBTENIDOS CON ESTE ESQUEMA DE MAPA ASOCIATIVO PARCIAL SUPERAN APROXIMADAMENTE EN UN 100 % A LOS RENDIMIENTOS OBTENIDOS CON ESQUEMAS DE MAPA ASOCIATIVO DE PAGINA COMPLETO.
- v UN PROGRAMA HACE REFERENCIA A LA DIRECCION VIRTUAL $V = (P,D)$.
- v EL MECANISMO DE TRADUCCION DE DIRECCIONES INTENTA ENCONTRAR LA PAGINA “P” EN EL MAPA DE PAGINA ASOCIATIVO PARCIAL:
 - u SI “P” SE ENCUENTRA ALLI:
 - ⊕ EL MAPA ASOCIATIVO DEVUELVE “P ‘ “ COMO EL N° DE MARCO DE PAGINA CORRESPONDIENTE A LA PAGINA VIRTUAL “P”.
 - ⊕ “P ‘ “ SE CONCATENA CON EL DESPLAZAMIENTO “D” PARA FORMAR LA DIRECCION REAL “R” QUE CORRESPONDE A LA DIRECCION VIRTUAL $V = (P,D)$.
 - u SI “P” NO SE ENCUENTRA EN EL MAPA DE PAGINA PARCIAL:
 - ⊕ SE UTILIZA UN MAPA DIRECTO CONVENCIONAL.
 - ⊕ LA DIRECCION “B” DEL REGISTRO DE ORIGEN DE LA TABLA DE PAGINAS SE AÑADE A “P” PARA LOCALIZAR LA ENTRADA APROPIADA A LA PAGINA “P” EN LA TABLA DE MAPA DE PAGINAS DE TRANSFORMACION DIRECTA DEL ALMACENAMIENTO PRIMARIO.

CONCEPTOS BASICOS DE PAGINACION

- ❖ LA TABLA INDICA QUE “P’ “ ES EL MARCO DE PAGINA CORRESPONDIENTE A LA PAGINA VIRTUAL “P”.
- ❖ “P’ “ SE CONCATENA CON EL DESPLAZAMIENTO “D” PARA FORMAR LA DIRECCION REAL “R” CORRESPONDIENTE A LA DIRECCION VIRTUAL $V = (P,D)$.

TRADUCCION DE DIRECCIONES DE PAGINACION POR COMBINACION DE TRANSFORMACION ASOCIATIVA / DIRECTA

REGISTRO ORIGEN DE LA TABLA DE PAGINA

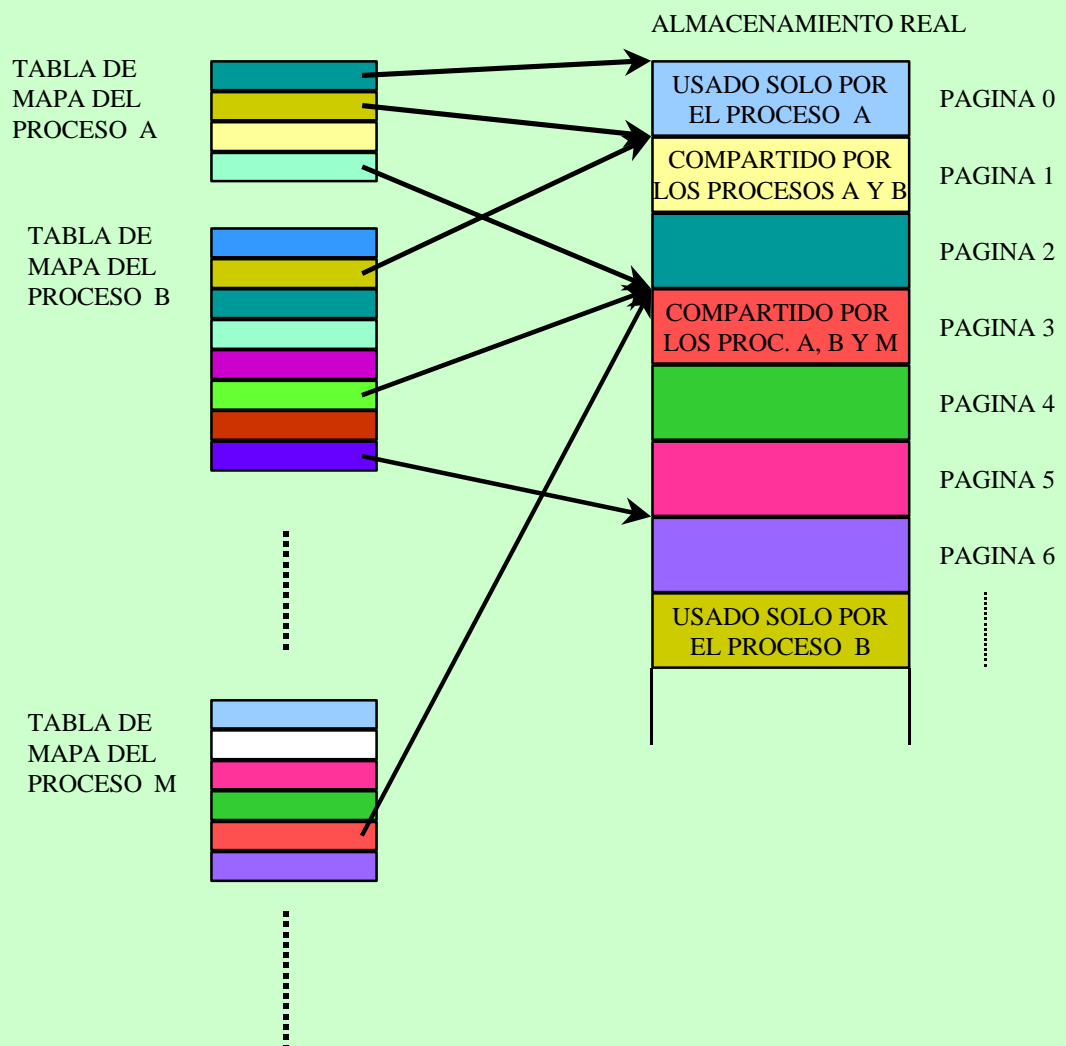


CONCEPTOS BASICOS DE PAGINACION

- v **COMPARTIMIENTO DE RECURSOS EN UN SISTEMA DE PAGINACION:**
- v EN SISTEMAS MULTIPROGRAMADOS, ESPECIALMENTE EN LOS DE TIEMPO COMPARTIDO, ES COMUN QUE MAS DE UN USUARIO ESTEN EJECUTANDO LOS MISMOS PROGRAMAS:
 - u PARA OPTIMIZAR EL USO DE LA MEMORIA REAL SE COMPARTEN LAS PAGINAS QUE PUEDEN SER COMPARTIDAS:
 - ⊕ EL COMPARTIMIENTO DEBE SER CUIDADOSAMENTE CONTROLADO PARA EVITAR QUE UN PROCESO MODIFIQUE DATOS QUE OTRO PROCESO ESTA LEYENDO.
 - ⊕ LOS PROGRAMAS SE ENCUENTRAN DIVIDIDOS EN AREAS SEPARADAS DE “PROCEDIMIENTO” Y “DATOS”.
 - ⊕ LOS PROCEDIMIENTOS NO MODIFICABLES SE LLAMAN “PROCEDIMIENTOS PUROS REENTRANTES”.
 - ⊕ LOS DATOS Y PROCEDIMIENTOS MODIFICABLES NO PUEDEN SER COMPARTIDOS.
 - ⊕ LOS DATOS NO MODIFICABLES (EJ.: TABLAS FIJAS) SON COMPARTIBLES.
 - u SE DEBE IDENTIFICAR CADA PAGINA COMO COMPARTIBLE O NO.
 - u HABRA MARCOS (CELDAS) DE PAGINAS COMPARTIDOS POR VARIOS PROCESOS.
- v EL COMPARTIMIENTO:
 - u REDUCE LA CANTIDAD DE ALMACENAMIENTO PRIMARIO NECESARIO PARA LA EJECUCION EFICAZ DE UN GRUPO DE PROCESOS.
 - u PUEDE HACER POSIBLE QUE UN SISTEMA DETERMINADO MANTENGA UNA CANTIDAD MAYOR DE USUARIOS (PROCESOS).

CONCEPTOS BASICOS DE PAGINACION

COMPARTIMIENTO EN UN SISTEMA DE PAGINACION PURA



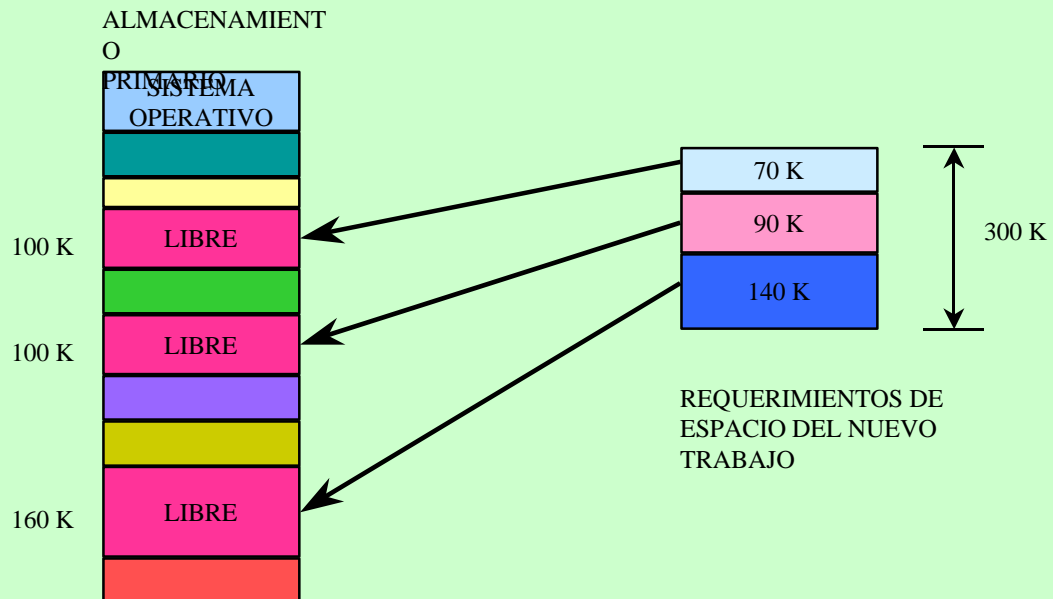
SEGMENTACION

- v EN LOS SISTEMAS DE “SEGMENTACION” UN PROGRAMA Y SUS DATOS PUEDEN OCUPAR VARIOS BLOQUES SEPARADOS DE ALMACENAMIENTO REAL.
- v LOS BLOQUES:
 - v NO NECESITAN SER DE IGUAL TAMAÑO.
 - v LOS BLOQUES SEPARADOS NO NECESITAN SER ADYACENTES.
 - v DEBEN ESTAR COMPUESTOS DE POSICIONES CONTIGUAS DE ALMACENAMIENTO.
- v SE COMPLICA LA PROTECCION DE BLOQUES DE MEMORIA DE UN PROCESO DE USUARIO.
- v ES MAS DIFICIL LIMITAR EL RANGO DE ACCESO DE CUALQUIER PROGRAMA.
- v UN ESQUEMA POSIBLE DE PROTECCION ES EL USO DE CLAVES DE PROTECCION DEL ALMACENAMIENTO:
 - v LAS CLAVES ESTAN BAJO EL CONTROL ESTRICTO DEL S. O.
 - v UN PROGRAMA DE USUARIO, A QUIEN CORRESPONDE UNA CIERTA CLAVE EN LA CPU, SOLO PUEDE HACER REFERENCIA A LOS OTROS BLOQUES DEL ALMACENAMIENTO CON IGUAL CLAVE DE PROTECCION.
- v UNA DIRECCION VIRTUAL ES UN PAR ORDENADO $V=(S,D)$:
 - v “S” ES EL N° DEL SEGMENTO DEL ALMACENAMIENTO VIRTUAL EN EL CUAL RESIDEN LOS ELEMENTOS REFERIDOS.
 - v “D” ES EL DESPLAZAMIENTO EN EL SEGMENTO “S” EN EL CUAL SE ENCUENTRA EL ELEMENTO REFERIDO.
- v UN PROCESO SOLO PUEDE EJECUTARSE SI SU SEGMENTO ACTUAL (COMO MINIMO) ESTA EN EL ALMACENAMIENTO PRIMARIO.
- v LOS SEGMENTOS SE TRANSFIEREN DEL ALMACENAMIENTO SECUNDARIO AL PRIMARIO COMO UNIDADES COMPLETAS.
- v UN NUEVO SEGMENTO PUEDE SER COLOCADO EN UNA SERIE DISPONIBLE DE POSICIONES CONTIGUAS DEL ALMACENAMIENTO PRIMARIO DE TAMAÑO SUFICIENTE PARA ALOJAR AL SEGMENTO.
- v LA TRADUCCION DINAMICA DE DIRECCIONES UTILIZA UNA “TABLA DE

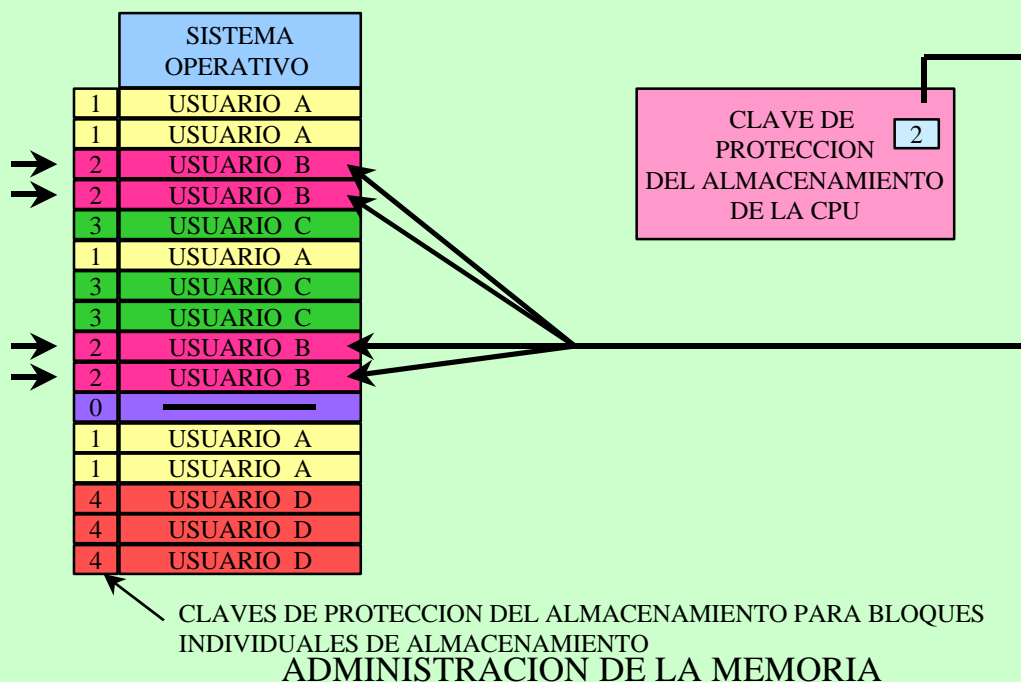
MAPA DE SEGMENTOS”.

SEGMENTACION

ASIGNACION NO CONTIGUA DE ALMACENAMIENTO



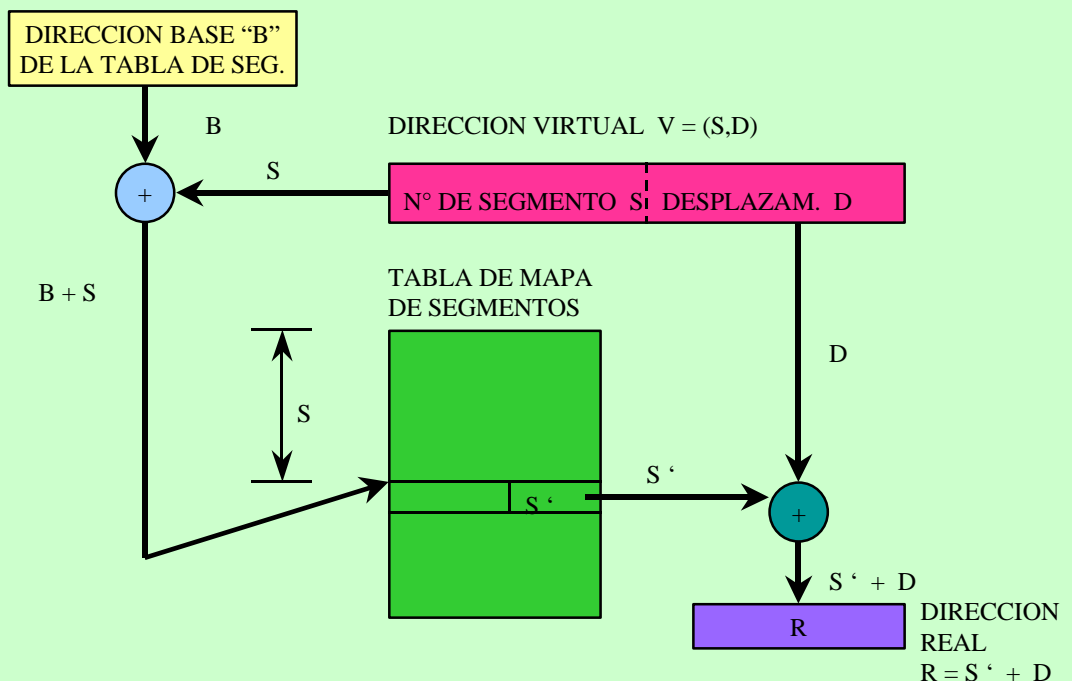
PROTECCION DEL ALMACENAMIENTO CON CLAVES EN SISTEMAS DE MULTIPROGRAMACION DE ASIGNACION NO CONTIGUA DE ALMACENAMIENTO



SEGMENTACION

TRADUCCION DE DIRECCION VIRTUAL EN UN SISTEMA DE SEGMENTACION PURA

REGISTRO ORIGEN DE LA TABLA DE MAPA DE SEGMENTOS



- v **CONTROL DE ACCESO EN SISTEMAS DE SEGMENTACION:**
- v SE LE OTORGA A CADA PROCESO CIERTOS DERECHOS DE ACCESO A TODOS LOS SEGMENTOS Y SE LE NIEGA COMPLETAMENTE EL ACCESO A MUCHOS OTROS.
- v SI UN PROCESO TIENE "ACCESO DE LECTURA" A UN SEGMENTO, PUEDE OBTENER CUALQUIER ELEMENTO DE INFORMACION CONTENIDO EN ESE SEGMENTO.
- v SI UN PROCESO TIENE "ACCESO DE ESCRITURA" A UN SEGMENTO, PUEDE MODIFICAR CUALQUIER CONTENIDO DEL SEGMENTO Y PUEDE INTRODUCIRLE INFORMACION ADICIONAL, INCLUSO DESTRUIR TODA LA INFORMACION DEL SEGMENTO.
- v UN PROCESO CON "ACCESO DE EJECUCION" DE UN SEGMENTO PUEDE EJECUTARLO COMO SI FUERA UN PROGRAMA.
- v UN PROCESO CON "ACCESO DE ADICION" PUEDE ESCRIBIR INFORMACION ADICIONAL AL FINAL DEL SEGMENTO, PERO NO PUEDE MODIFICAR LA INFORMACION EXISTENTE.

SEGMENTACION

- v EN BASE A LOS “TIPOS DE CONTROL DE ACCESO” INDICADOS PUEDEN CREARSE DISTINTOS “MODOS DE CONTROL DE ACCESO”.
- v EJ. DE COMBINACION DE LOS ACCESOS DE LECTURA, ESCRITURA Y EJECUCION PARA PRODUCIR MODOS DE PROTECCION UTILES:
- v MODO LECT. ESCR. EJECUC. EXPLICACION APLICACION

0	N	N	N	NO HAY PERMISO DE ACCESO	SEGURIDAD
1	N	N	S	SOLO EJECUCION	UN PROGR. DISPONIBLE A LOS USUARIOS, QUE NO PUEDEN COPIARLO NI MODIFICARLO, PERO SI EJECUTARLO
2	S	N	N	SOLO LECTURA	RECUPERACION DE INFORMAC.
3	S	N	S	LECTURA / EJECUCION	UN PROGRAMA PUEDE SER COPIADO O EJECUTADO, PERO
4	S	S	N	LECTURA / ESCRITURA PERO NO EJECUCION	NO PUEDE SER MODIFICADO UN INTENTO ERRONEO DE EJECUTARLOS
5	S	S	S	ACCESO NO LIMITADO	ESTE ACCESO SE CONCEDE A LOS USUARIOS DE CONFIANZA

SEGMENTACION

- v **TRADUCCION DE DIRECCIONES DE SEGMENTACION POR TRANSFORMACION DIRECTA:**
- v EXISTEN VARIAS ESTRATEGIAS PARA LA IMPLEMENTACION DE LA TRADUCCION DE DIRECCIONES DE SEGMENTACION:
 - v POR TRANSFORMACION DIRECTA, ASOCIATIVA O COMBINACION DE ASOCIATIVA / DIRECTA.
 - v CON CACHE SUFICIENTE PARA ALOJAR LA TABLA COMPLETA DE MAPA DE SEGMENTOS O CACHE PARCIALES QUE CONTENGAN SOLO LAS ENTRADAS DE LOS SEGMENTOS DE REFERENCIA MAS RECIENTE.
- v SE CONSIDERARA LA TRADUCCION DE DIRECCIONES DE SEGMENTACION CON LA TABLA COMPLETA DE MAPA DE SEGMENTOS EN LA CACHE.
- v UN PROCESO EN EJECUCION HACE REFERENCIA A LA DIRECCION VIRTUAL $V = (S,D)$:
 - v EL SEGMENTO N° “S” SE AÑADE A LA DIRECCION BASE “B” EN EL REGISTRO ORIGEN DE LA TABLA DE MAPA DE SEGMENTOS FORMANDO LA DIRECCION DE MEMORIA REAL “B + S”, DE LA ENTRADA PARA EL SEGMENTO “S” DE LA TABLA DE MAPA DE SEGMENTOS:
 - Φ LA TABLA CONTIENE LA DIRECCION DEL ALMACENAMIENTO PRIMARIO “S ‘ “, DONDE COMIENZA EL SEGMENTO.
 - v EL DESPLAZAMIENTO “D” SE AÑADE A “S ‘ “ FORMANDO LA DIRECCION REAL “R = D + S ‘ “, CORRESPONDIENTE A LA DIRECCION VIRTUAL “V = (S,D)”.
- v UN “BIT DE RESIDENCIA”, “R”, INDICA SI EN LA ACTUALIDAD EL SEGMENTO SE ENCUENTRA O NO EN EL ALMACENAMIENTO PRIMARIO.
- v SI EL SEGMENTO SE ENCUENTRA EN EL ALMACENAMIENTO PRIMARIO:
 - v “S ‘ “ ES LA DIRECCION EN ESTE ALMACENAMIENTO DONDE COMIENZA EL SEGMENTO.
- v SI EL SEGMENTO NO SE ENCUENTRA EN EL ALMACENAMIENTO PRIMARIO:
 - v “A” ES LA DIRECCION EN EL ALMACENAMIENTO SECUNDARIO DE DONDE DEBE RECUPERARSE ANTES QUE EL PROCESO PUEDA

SEGMENTACION

- v SE COMPARA CADA REFERENCIA A UN SEGMENTO CON LOS BITS DE PROTECCION PARA DETERMINAR SI SE PERMITE LA OPERACION QUE SE ESTA INTENTANDO.
- v SI EL SEGMENTO BUSCADO NO ESTA EN EL ALMACENAMIENTO PRIMARIO SE GENERA UN “FALLO DE PERDIDA DE SEGMENTO”:
 - v EL S. O. OBTIENE EL CONTROL Y CARGA EL SEGMENTO REFERIDO DESDE LA DIRECCION “A” DEL ALMACENAMIENTO SECUNDARIO.
 - v SE COMPRUEBA SI EL DESPLAZAMIENTO “D” ES MENOR O IGUAL A LA LONGITUD DEL SEGMENTO “L”:
 - ⊕ SI NO ES ASI SE GENERA UN “FALLO DE DESBORDAMIENTO DE SEGMENTO”:
 - EL S. O. OBTIENE EL CONTROL Y TERMINA LA EJECUCION DEL PROCESO.
 - ⊕ SI EL DESPLAZAMIENTO ESTA EN EL RANGO DEL SEGMENTO SE COMPRUEBAN LOS BITS DE PROTECCION PARA ASEGURARSE SI SE PERMITE LA OPERACION QUE SE ESTA INTENTANDO:
 - SI ES ASI ENTONCES LA DIRECCION BASE DEL SEGMENTO, “S ‘”, EN EL ALMACENAMIENTO PRIMARIO SE AÑADE AL DESPLAZAMIENTO “D” FORMANDO LA DIRECCION DE MEMORIA REAL “ $R = S' + D$ ”, QUE CORRESPONDE A LA DIRECCION DEL ALMACENAMIENTO VIRTUAL “ $V = (S,D)$ ”.
 - SI LA OPERACION INTENTADA NO SE PERMITE SE GENERA UN “FALLO DE PROTECCION DE SEGMENTO”:
 - EL S. O. OBTIENE EL CONTROL Y TERMINA LA EJECUCION DEL PROCESO.

SEGMENTACION

- v **COMPARTIMIENTO EN UN SISTEMA DE SEGMENTACION:**
- v UNA DE LAS VENTAJAS DE LA SEGMENTACION SOBRE LA PAGINACION ES QUE SE TRATA MAS DE UN HECHO LOGICO QUE FISICO:
 - u EN UN SISTEMA DE SEGMENTACION, UNA VEZ QUE UN SEGMENTO HA SIDO DECLARADO COMO COMPARTIDO, ENTONCES LAS ESTRUCTURAS QUE LO INTEGRAN PUEDEN CAMBIAR DE TAMAÑO:
 - φ NO CAMBIA EL HECHO LOGICO DE QUE RESIDEN EN UN SEGMENTO COMPARTIDO.
- v DOS PROCESOS PUEDEN COMPARTIR UN SEGMENTO CON SOLO TENER ENTRADAS EN SUS TABLAS GENERALES QUE APUNTEN AL MISMO SEGMENTO DEL ALMACENAMIENTO PRIMARIO.

ENTRADA DE TABLA DE MAPA DE SEGMENTOS

	DIRECCION DE ALMAC. SECUN. (SI EL SEGM. NO ESTA EN EL ALMAC. REAL)	LONGITUD DEL SEGM.	BITS DE PROTECCION	DIRECCION BASE DEL SEGM. (SI EL SEGM. ESTA EN EL ALMACEN.)
BIT DE RESIDENCIA DEL SEGM.				
R	A	L	R W E A	S

R = 0 SI EL SEGMENTO NO ESTA EN EL ALMACENAMIENTO PRIMARIO

R = 1 SI EL SEGMENTO ESTA EN EL ALMACENAMIENTO PRIMARIO

BITS DE PROTECCION: (1-SI, 0-NO)

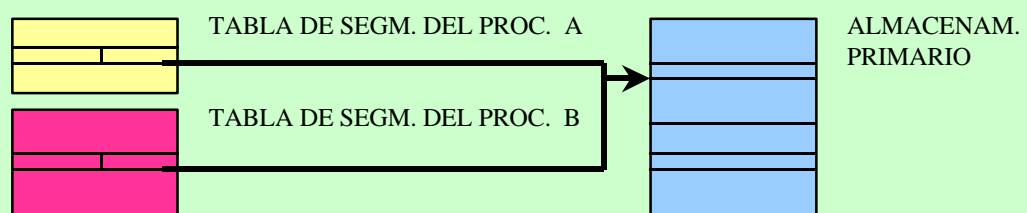
R - ACCESO DE LECTURA

E - ACCESO DE EJECUCION

W - ACCESO DE ESCRITURA

A - ACCESO DE ADICION

COMPARTIMIENTO EN UN SISTEMA DE SEGMENTACION PURA



ADMINISTRACION DE LA MEMORIA

SISTEMAS DE PAGINACION / SEGMENTACION

- v OFRECEN LAS VENTAJAS DE LAS DOS TECNICAS DE ORGANIZACION DEL ALMACENAMIENTO VIRTUAL.
 - v EL TAMAÑO DE LOS SEGMENTOS ES MULTIPLO DEL DE LAS PAGINAS.
 - v NO ES NECESARIO QUE TODAS LAS PAGINAS DE UN SEGMENTO SE ENCUENTREN AL MISMO TIEMPO EN EL ALMACENAMIENTO PRIMARIO.
 - v LAS PAGINAS DE ALMACENAMIENTO VIRTUAL, QUE SON CONTIGUAS EN ESTE ALMACENAMIENTO, NO NECESITAN SER CONTIGUAS EN EL ALMACENAMIENTO REAL.
 - v EL DIRECCIONAMIENTO ES TRIDIMENSIONAL CON UNA DIRECCION DE ALMACENAMIENTO VIRTUAL “V = (S,P,D)”:
 - u “S” ES EL N° DEL SEGMENTO; “P” ES EL N° DE PAGINA Y “D” ES EL DESPLAZAMIENTO EN LA PAGINA DONDE SE ENCUENTRA ASIGNADO EL ELEMENTO DESEADO.
 - v **TRADUCCION DINAMICA DE DIRECCIONES EN SISTEMAS DE PAGINACION / SEGMENTACION:**
 - v SE CONSIDERA LA TRADUCCION DINAMICA DE DIRECCIONES DE VIRTUALES A REALES EN UN SISTEMA DE PAGINACION / SEGMENTACION UTILIZANDO LA COMBINACION DE TRANSFORMACION ASOCIATIVA / DIRECTA.
 - v EL PROCESO EN EJECUCION HACE REFERENCIA A LA DIRECCION VIRTUAL $V = (S,P,D)$.
 - v LAS PAGINAS DE REFERENCIA MAS RECIENTE TIENEN ENTRADAS EN UN ALMACENAMIENTO ASOCIATIVO.
 - v SE REALIZA UNA BUSQUEDA ASOCIATIVA PARA INTENTAR LOCALIZAR (S,P) EN EL ALMACENAMIENTO ASOCIATIVO:
 - u SI SE ENCUENTRA (S,P), ENTONCES EL MARCO DE PAGINA “P” EN EL CUAL RESIDE DICHA PAGINA EN LA MEMORIA REAL, SE CONCATENA AL DESPLAZAMIENTO “D” PARA FORMAR LA DIRECCION DE MEMORIA REAL “R” CORRESPONDIENTE A LA DIRECCION VIRTUAL $V = (S,P,D)$.
 - u SI NO SE ENCUENTRA (S,P), ENTONCES:
 - LA DIRECCION BASE “B” DE LA TABLA DE SEGMENTOS SE AÑADE AL N° DE SEGMENTO “S” FORMANDO LA DIRECCION “B + S” DE LA ENTRADA DE LA TABLA DE MAPA DE SEGMENTOS PARA EL SEGMENTO “S” DE LA MEMORIA REAL.
- ADMINISTRACION DE LA MEMORIA 127

SISTEMAS DE PAGINACION / SEGMENTACION

- LA ENTRADA DE LA TABLA DE MAPA DE SEGMENTOS INDICA LA DIRECCION BASE “S ‘ “ DE LA TABLA DE PAGINAS PARA EL SEGMENTO “S”.
- EL N° DE PAGINA “P” SE AÑADE A “S ‘ “ FORMANDO LA DIRECCION “P + S ‘ “ DE LA ENTRADA EN LA TABLA DE PAGINAS PARA LA PAGINA “P” DEL SEGMENTO “S”:
 - INDICA QUE “P ‘ “ ES EL N° DEL MARCO CORRESPONDIENTE A LA PAGINA VIRTUAL “P”.
 - “P ‘ “ SE CONCATENA CON EL DESPLAZAMIENTO “D” FORMANDO LA DIRECCION REAL “R” QUE CORRESPONDE A LA DIRECCION VIRTUAL $V = (S,P,D)$.
- v SI EL SEGMENTO “S” NO SE ENCUENTRA EN EL ALMACENAMIENTO PRIMARIO SE PRODUCE UN “FALLO DE PERDIDA DE SEGMENTO”:
 - EL S. O. LOCALIZA EL SEGMENTO EN EL ALMACENAMIENTO SECUNDARIO, CREA UNA TABLA DE PAGINAS PARA EL SEGMENTO Y CARGA LA PAGINA APROPIADA EN EL ALMACENAMIENTO PRIMARIO, PUDIENDO PRODUCIR REEMPLAZOS DE PAGINAS.
- v SI EL SEGMENTO “S” ESTA EN EL ALMACENAMIENTO PRIMARIO Y SI LA REFERENCIA A LA TABLA DE MAPA DE PAGINAS INDICA QUE LA PAGINA DESEADA NO SE ENCUENTRA EN EL ALMACENAMIENTO PRIMARIO SE PRODUCE UN “FALLO DE PERDIDA DE PAGINA”:
 - EL S. O. OBTIENE EL CONTROL, LOCALIZA LA PAGINA EN EL ALMACENAMIENTO SECUNDARIO Y LA CARGA, PUDIENDO REEMPLAZAR OTRA PAGINA.
- v SI UNA DIRECCION DE ALMACENAMIENTO VIRTUAL ESTA MAS ALLA DEL FINAL DEL SEGMENTO SE GENERA UN “FALLO DE DESBORDAMIENTO DE SEGMENTO”; DEBE SER ATENDIDO POR EL S. O.
- v SI LOS BITS DE PROTECCION INDICAN QUE LA OPERACION QUE SE VA A EJECUTAR EN LA DIRECCION VIRTUAL REFERIDA NO SE PERMITE SE GENERA UN “FALLO DE PROTECCION DE SEGMENTO”; DEBE SER ATENDIDO POR EL S. O.
- v SI SE UTILIZA UN MECANISMO DE TRANSFORMACION DIRECTA PURA, MANTENIENDO EL MAPA COMPLETO DENTRO DEL ALMACENAMIENTO PRIMARIO, LA REFERENCIA PROMEDIO DE ALMACENAMIENTO VIRTUAL

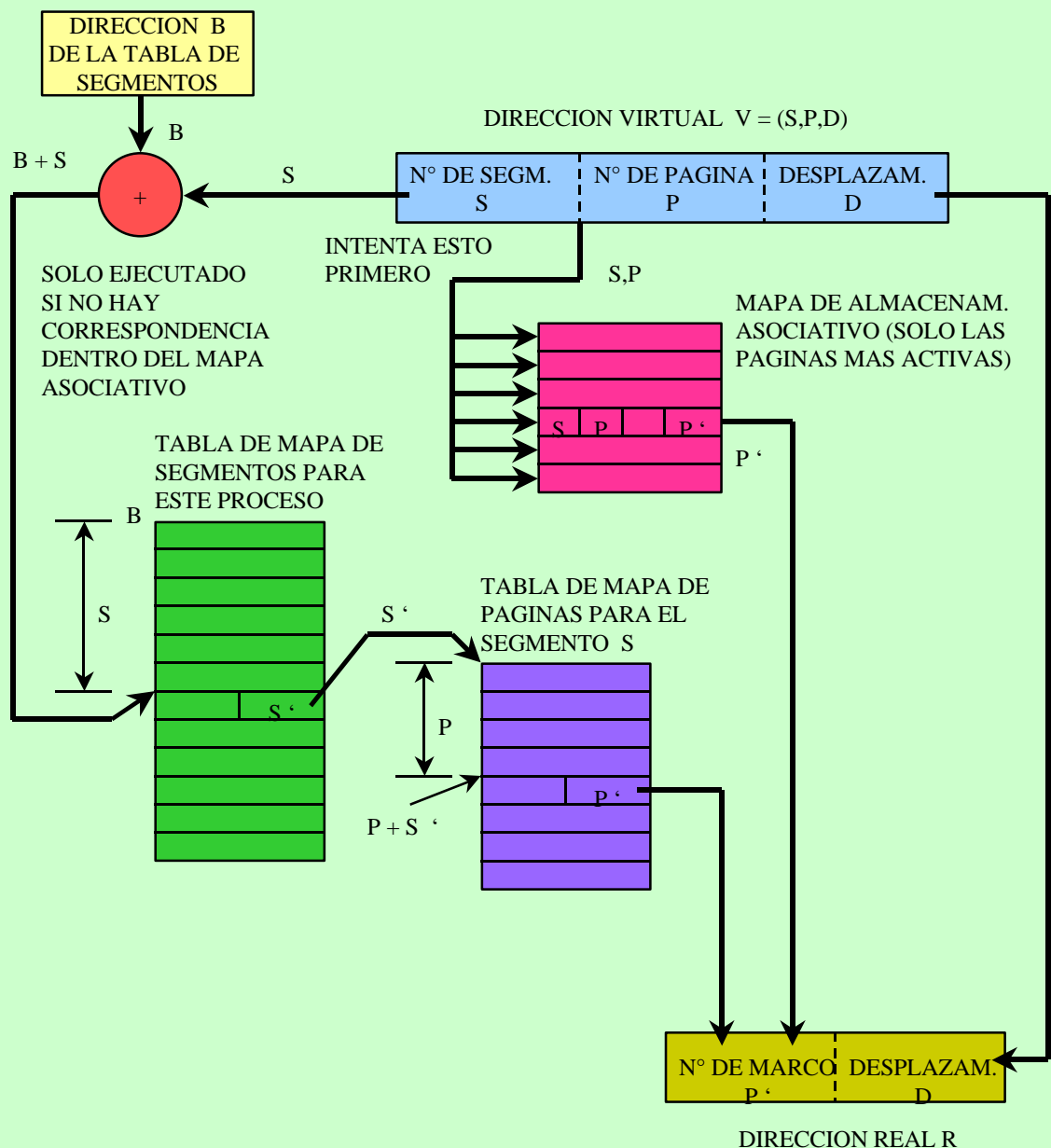
SISTEMAS DE PAGINACION / SEGMENTACION

- v UN CICLO DE ALMACENAMIENTO PARA ACCEDER A LA TABLA DE MAPA DE SEGMENTOS.
- v UN SEGUNDO CICLO DE ALMACENAMIENTO PARA HACER REFERENCIA A LA TABLA DE MAPA DE PAGINAS.
- v UN TERCER CICLO DE ALMACENAMIENTO PARA REFERENCIAR AL ELEMENTO DESEADO DEL ALMACENAMIENTO REAL.
- v CADA REFERENCIA A UN ELEMENTO COMPRENDE TRES CICLOS DE ALMACENAMIENTO:
 - v EL SISTEMA CORRERIA CASI A 1 / 3 DE SU VELOCIDAD NOMINAL.
 - v LA TRADUCCION DE DIRECCIONES INSUMIRIA 2 / 3 DEL TIEMPO.
- v CON LA UTILIZACION DE REGISTROS ASOCIATIVOS (POR EJ. 16 REG.) SE LOGRAN VELOCIDADES DE EJECUCION DEL 90 % O MAS DE LA VELOCIDAD TOTAL DE PROCESAMIENTO DE SUS PROCESADORES DE CONTROL.
- v LA ESTRUCTURA DE TABLAS DE PROCESOS, DE MAPAS DE SEGMENTOS Y DE MAPAS DE PAGINAS PUEDE CONSUMIR UN PORCENTAJE IMPORTANTE DEL ALMACENAMIENTO PRIMARIO CUANDO SE EJECUTAN UN GRAN N° DE PROCESOS.
- v LA TRADUCCION PROCEDE MUCHO MAS RAPIDO SI TODAS LAS TABLAS ESTAN EN EL ALMACENAMIENTO PRIMARIO, LO QUE RESTA ESPACIO PARA LOS PROCESOS.
- v **COMPARTIMIENTO EN UN SISTEMA DE PAGINACION / SEGMENTACION:**
- v SE IMPLEMENTA DISPONIENDO ENTRADAS EN TABLAS DE MAPA DE SEGMENTOS PARA DIFERENTES PROCESOS QUE APUNTEN A LA MISMA TABLA DE MAPA DE PAGINAS.
- v EL COMPARTIMIENTO REQUIERE UNA ADMINISTRACION CUIDADOSA POR PARTE DEL S. O., YA SEA EN SISTEMAS DE PAGINACION, SEGMENTACION O PAGINACION / SEGMENTACION:
 - v SE DEBE CONSIDERAR QUE SUCEDERIA SI UNA NUEVA PAGINA REEMPLAZARA A OTRA PAGINA COMPARTIDA POR MUCHOS PROCESOS.

SISTEMAS DE PAGINACION / SEGMENTACION

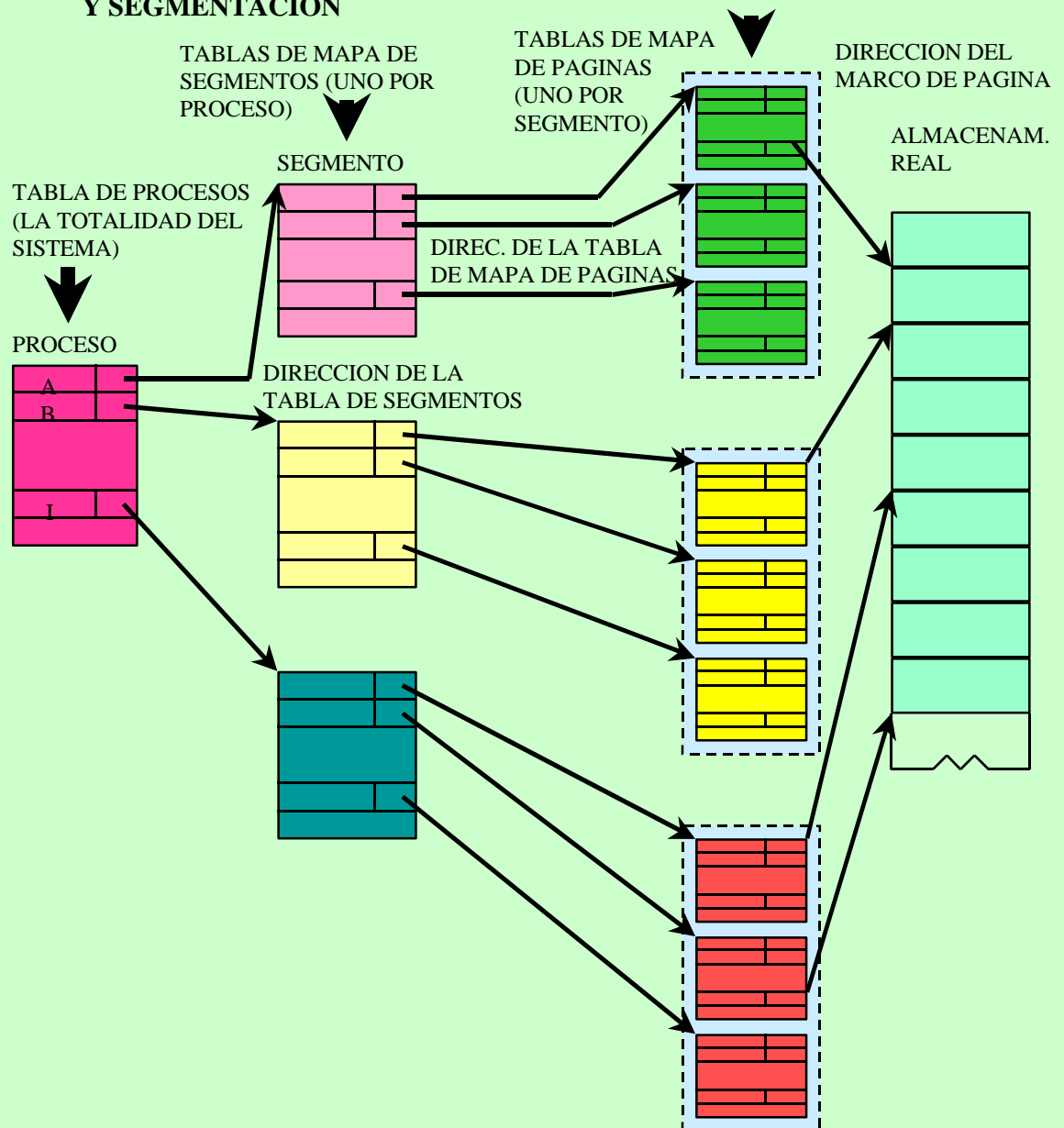
TRADUCCION DE DIRECCIONES VIRTUALES CON COMBINACION DE TRANSFORMACION ASOCIATIVA / DIRECTA DENTRO DE UN SISTEMA DE PAGINACION Y SEGMENTACION

REGISTRO ORIGEN DE LA TABLA DE SEGMENTOS



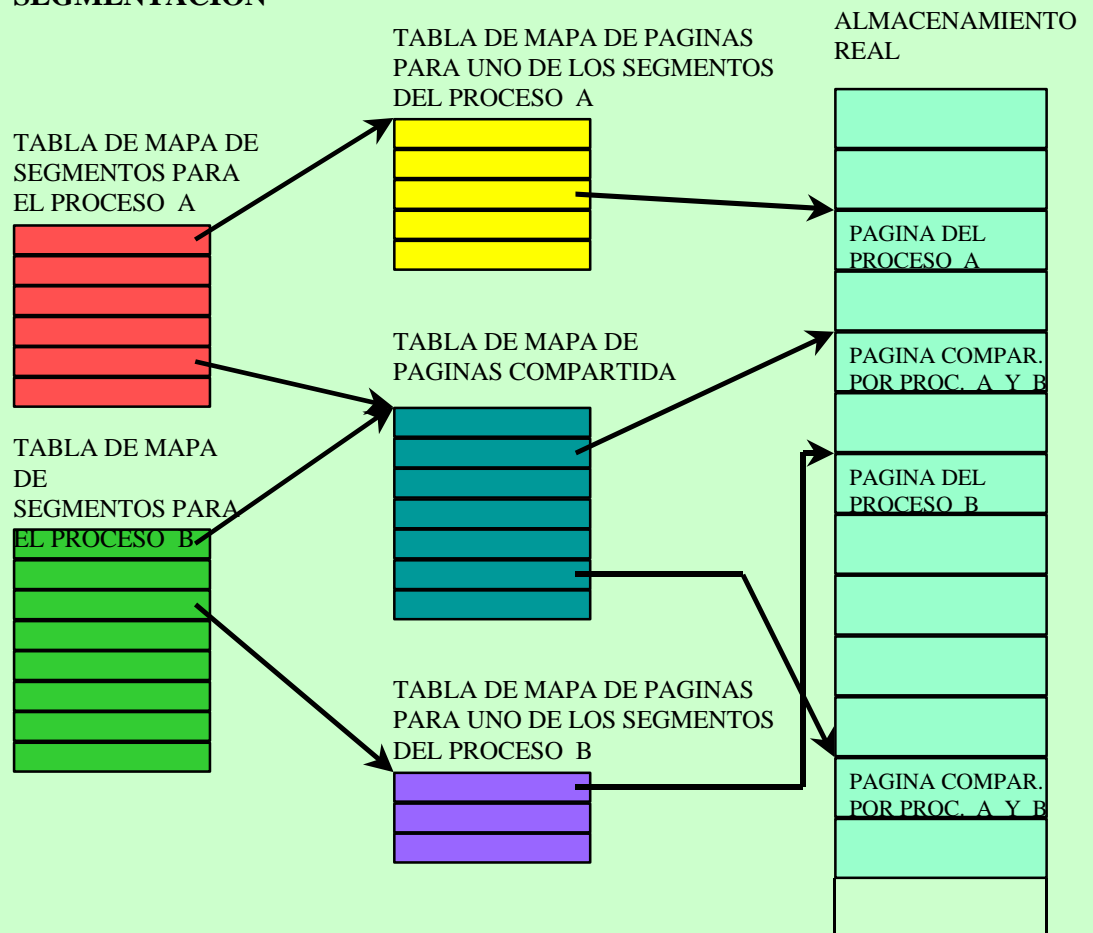
SISTEMAS DE PAGINACION / SEGMENTACION

ESTRUCTURA DE TABLAS PARA UN SISTEMA DE PAGINACION Y SEGMENTACION



SISTEMAS DE PAGINACION / SEGMENTACION

DOS PROCESOS COMPARTIENDO UN SISTEMA DE PAGINACION Y SEGMENTACION



INTRODUCCION A LA ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

- v LAS DIFERENTES ORGANIZACIONES DE ALMACENAMIENTO VIRTUAL GENERALMENTE IMPLEMENTADAS SON:
 - v PAGINACION, SEGMENTACION Y SEGMENTACION Y PAGINACION.
- v LAS ESTRATEGIAS PARA LA ADMINISTRACION DE SISTEMAS DE ALMACENAMIENTO VIRTUAL CONDICIONAN LA CONDUCTA DE LOS SISTEMAS DE ALMACENAMIENTO VIRTUAL QUE OPERAN SEGUN ESAS ESTRATEGIAS.
- v “ESTRATEGIAS DE BUSQUEDA”:
 - v TRATAN DE LOS CASOS EN QUE UNA PAGINA O SEGMENTO DEBEN SER TRAIDOS DEL ALMACENAMIENTO SECUNDARIO AL PRIMARIO.
 - v LAS ESTRATEGIAS DE “BUSQUEDA POR DEMANDA” ESPERAN A QUE SE HAGA REFERENCIA A UNA PAGINA O SEGMENTO POR UN PROCESO ANTES DE TRAERLOS AL ALMACENAMIENTO PRIMARIO.
 - v LOS ESQUEMAS DE “BUSQUEDA ANTICIPADA” INTENTAN DETERMINAR POR ADELANTADO A QUE PAGINAS O SEGMENTOS HARA REFERENCIA UN PROCESO PARA TRAERLOS AL ALMACENAMIENTO PRIMARIO ANTES DE SER EXPLICITAMENTE REFERENCIADOS.
- v “ESTRATEGIAS DE COLOCACION”:
 - v TRATAN DEL LUGAR DEL ALMACENAMIENTO PRIMARIO DONDE SE COLOCARA UNA NUEVA PAGINA O SEGMENTO.
 - v LOS SISTEMAS TOMAN LAS DECISIONES DE COLOCACION DE UNA FORMA TRIVIAL YA QUE UNA NUEVA PAGINA PUEDE SER COLOCADA DENTRO DE CUALQUIER MARCO DE PAGINA DISPONIBLE.
- v “ESTRATEGIAS DE REPOSICION”:
 - v TRATAN DE LA DECISION DE CUAL PAGINA O SEGMENTO DESPLAZAR PARA HACER SITIO A UNA NUEVA PAGINA O SEGMENTO CUANDO EL ALMACENAMIENTO PRIMARIO ESTA COMPLETAMENTE COMPROMETIDO.

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

✓ **ESTRATEGIAS DE REPOSICION DE PAGINA:**

✓ LAS PRINCIPALES SON:

- ✓ EL PRINCIPIO DE OPTIMIZACION.
- ✓ REPOSICION DE PAGINAS AL AZAR.
- ✓ PRIMERO EN ENTRAR - PRIMERO EN SALIR.
- ✓ MENOS RECIENTEMENTE USADA.
- ✓ MENOS FRECUENTEMENTE USADA.
- ✓ NO USADA RECIENTEMENTE.
- ✓ CONJUNTOS DE TRABAJO.

✓ **EL PRINCIPIO DE OPTIMIZACION:**

- ✓ EL “PRINCIPIO DE OPTIMIZACION” INDICA QUE PARA OBTENER UN RENDIMIENTO OPTIMO, LA PAGINA QUE SE VA A REPONER ES UNA QUE NO SE VA A UTILIZAR EN EL FUTURO DURANTE EL PERIODO DE TIEMPO MAS LARGO.

- ✓ EL PROBLEMA ES QUE NO ES FACTIBLE PREDECIR EL FUTURO.

✓ **REPOSICION DE PAGINA AL AZAR:**

- ✓ CONSISTE EN ESCOGER AL AZAR LA PAGINA QUE VA A SER REEMPLAZADA.

- ✓ TODAS LAS PAGINAS DEL ALMACENAMIENTO PRINCIPAL DEBEN TENER LA MISMA PROBABILIDAD DE SER REEMPLAZADAS.

- ✓ DEBE PODER SELECCIONAR CUALQUIER PAGINA, INCLUYENDO LA QUE VA A SER REFERENCIADA A CONTINUACION (PEOR SELECCION).

- ✓ ESTE ESQUEMA ES RARAMENTE USADO.

✓ **REPOSICION DE PAGINA POR EL SISTEMA DE PRIMERO EN ENTRAR - PRIMERO EN SALIR (FIFO):**

- ✓ SE REGISTRA EL MOMENTO EN QUE CADA PAGINA INGRESA AL ALMACENAMIENTO PRIMARIO.

- ✓ PARA REEMPLAZAR UNA PAGINA, SE SELECCIONA AQUELLA QUE HA ESTADO MAS TIEMPO ALMACENADA.

- ✓ SE PRESENTA EL INCONVENIENTE DE QUE SE PUEDEN REEMPLAZAR PAGINAS MUY USADAS:

- ✓ SERAN LLAMADAS DE NUEVO AL ALMACENAMIENTO PRIMARIO

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

v “ANOMALIA FIFO”:

- BELADY, NELSON Y SHEDLER DESCUBRIERON QUE CON LA REPOSICION FIFO, CIERTOS PATRONES DE REFERENCIAS DE PAGINAS CAUSAN MAS FALLOS DE PAGINAS CUANDO SE AUMENTA EL N° DE MARCOS (CELDA) DE PAGINAS ASIGNADOS A UN PROCESO: EN ESTO CONSISTE LA “ANOMALIA FIFO”.
- ESTA ANOMALIA CONTRADICE A LA INTUICION.

EJEMPLO DE ANOMALIA FIFO

REFERENCIAS DE PAGINAS REPOSICION CON TRES MARCOS DISPONIBLES

A FALLO	A	-	-
B FALLO	B	A	-
C FALLO	C	B	A
D FALLO	D	C	B
A FALLO	A	D	C
B FALLO	B	A	D
E FALLO	E	B	A
A <u>NO FALLO</u>	E	B	A
B <u>NO FALLO</u>	E	B	A
C FALLO	C	E	B
D FALLO	D	C	E
E <u>NO FALLO</u>	D	C	E

TRES “NO FALLOS”

REFERENCIAS DE PAGINAS REPOSICION CON CUATRO MARCOS DISPONIBLES

A FALLO	A	-	-	-
B FALLO	B	A	-	-
C FALLO	C	B	A	-
D FALLO	D	C	B	A
A <u>NO FALLO</u>	D	C	B	A
B <u>NO FALLO</u>	D	C	B	A
E FALLO	E	D	C	B
A FALLO	A	E	D	C
B FALLO	B	A	E	D
C FALLO	C	B	A	E
D FALLO	D	C	B	A
E FALLO	E	D	C	B

DOS “NO FALLOS”

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

- v **REPOSICION DE PAGINA MENOS - RECIENTEMENTE - USADA (LRU):**
- v ESTA ESTRATEGIA SELECCIONA PARA SER REEMPLAZADA LA PAGINA QUE NO HA SIDO USADA DURANTE EL MAYOR PERIODO DE TIEMPO.
- v SE BASA EN LA HEURISTICA DE QUE EL PASADO RECIENTE ES UN BUEN INDICADOR DEL FUTURO PROXIMO.
- v REQUIERE QUE CADA PAGINA RECIBA UN “SELLO DE TIEMPO” CADA VEZ QUE SE REFERENCIA:
 - u PUEDE SIGNIFICAR UNA SOBRECARGA ADICIONAL IMPORTANTE.
 - u NO SE IMPLEMENTA FRECUENTEMENTE.
- v LA PAGINA SELECCIONADA PARA REEMPLAZO PODRIA SER LA PROXIMA EN SER REQUERIDA:
 - u HABRIA QUE PAGINARLA DE NUEVO AL ALMACENAMIENTO PRINCIPAL CASI DE INMEDIATO.
- v **REPOSICION DE PAGINA MENOS - FRECUENTEMENTE - USADA (LFU):**
- v ACA INTERESA LA INTENSIDAD DE USO QUE HAYA TENIDO CADA PAGINA.
- v LA PAGINA QUE SERA REEMPLAZADA ES AQUELLA QUE HA SIDO USADA CON MENOS FRECUENCIA O QUE HA SIDO REFERIDA CON MENOS INTENSIDAD.
- v EL INCONVENIENTE ES QUE SE PUEDE SELECCIONAR FACILMENTE PARA SU REPOSICION LA PAGINA EQUIVOCADA:
 - u EJ.: LA PAGINA DE USO MENOS FRECUENTE PUEDE SER LA PAGINA DE ENTRADA MAS RECIENTE AL ALMACENAMIENTO PRINCIPAL:
 - ⦿ EXISTE UNA ALTA PROBABILIDAD DE QUE SEA USADA DE INMEDIATO.
- v **REPOSICION DE PAGINA NO USADA - RECIENTEMENTE (NUR):**
- v PRESUPONE QUE LAS PAGINAS QUE NO HAN TENIDO USO RECIENTE TIENEN POCA PROBABILIDAD DE SER USADAS EN EL FUTURO PROXIMO Y PUEDEN SER REEMPLAZADAS POR OTRAS NUEVAS.
- v ES DESEABLE REEMPLAZAR UNA PAGINA QUE NO HA SIDO CAMBIADA MIENTRAS ESTABA EN EL ALMACENAMIENTO PRIMARIO.

ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

- v LA ESTRATEGIA NUR SE IMPLEMENTA CON LA ADICION DE DOS BITS DE HARDWARE POR PAGINA:
 - υ A) “BIT REFERENCIADO”:
 - Φ = 0 SI LA PAGINA NO HA SIDO REFERENCIADA.
 - Φ = 1 SI LA PAGINA HA SIDO REFERENCIADA.
 - υ B) “BIT MODIFICADO” (TAMBIEN LLAMADO “BIT SUCIO”):
 - Φ = 0 SI LA PAGINA NO HA SIDO MODIFICADA.
 - Φ = 1 SI LA PAGINA HA SIDO MODIFICADA.
- v LA SELECCION DE LA PAGINA QUE SERA REEMPLAZADA COMIENZA BUSCANDO UNA PAGINA QUE NO HA SIDO REFERENCIADA:
 - υ SI NO LA ENCUENTRA HABRA QUE REEMPLAZAR UNA PAGINA QUE HA SIDO REFERENCIADA.
- v SI UNA PAGINA NO HA SIDO REFERENCIADA SE COMPRUEBA SI HA SIDO MODIFICADA O NO:
 - υ SI NO HA SIDO MODIFICADA SE LA REEMPLAZA:
 - Φ SU REPOSICION REPRESENTA MENOS SOBRECARGA QUE LA DE UNA PAGINA MODIFICADA:
 - DEBERIA GRABARSE DE NUEVO EN EL ALMACENAMIENTOS SECUNDARIO.
 - υ SI NO SE ENCUENTRA UNA PAGINA QUE NO HA SIDO MODIFICADA SERA REEMPLAZADA UNA PAGINA MODIFICADA.
- v CON EL TRANSCURSO DEL TIEMPO LA MAYORIA DE LOS “BITS REFERENCIADOS” SERAN ACTIVADOS:
 - υ SE PIERDE LA CAPACIDAD PARA DISTINGUIR LAS PAGINAS MAS DESEABLES PARA SER REEMPLAZADAS.
 - υ PARA EVITARLO SE AJUSTAN PERIODICAMENTE TODOS LOS “BITS REFERENCIADOS” A “0”:
 - Φ SE LOGRA UN NUEVO INICIO.
 - Φ SE VUELVE VULNERABLE AL REEMPLAZO AUN A LAS PAGINAS ACTIVAS, PERO SOLO BREVEMENTE, MIENTRAS SE REAJUSTAN LOS BITS.
- v LOS “BITS MODIFICADOS” NO SE AJUSTAN PERIODICAMENTE SEGUN

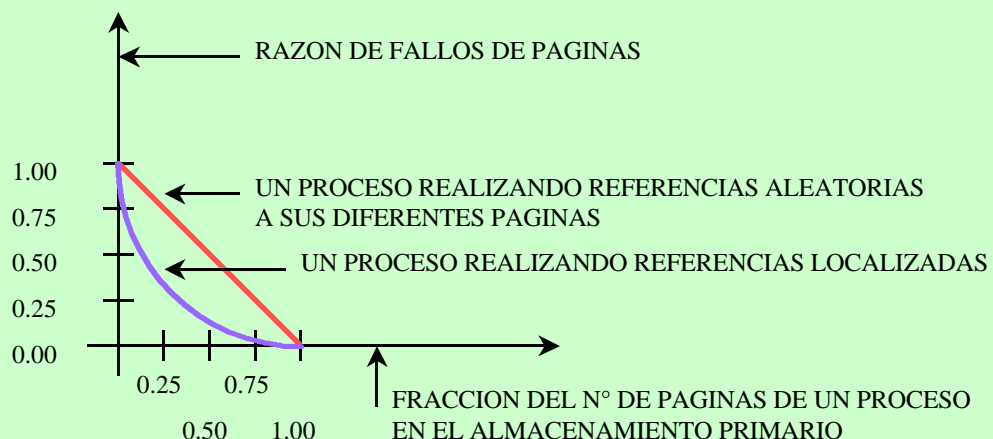
LOCALIDAD

- v CONCEPTO DE “LOCALIDAD”:
 - u “LOS PROCESOS TIENDEN A HACER REFERENCIA AL ALMACENAMIENTO EN PATRONES NO UNIFORMES Y MUY LOCALIZADOS”.
- v LA “LOCALIDAD” SE MANIFIESTA EN EL “TIEMPO” Y EN EL “ESPACIO”:
 - u ES UNA PROPIEDAD EMPIRICA (OBSERVADA).
 - u NUNCA ESTA GARANTIZADA PERO ES ALTAMENTE PROBABLE.
 - u EJ.: LOS PROCESOS TIENDEN A FAVORECER CIERTOS SUBCONJUNTOS DE PAGINAS, LAS QUE TIENDEN A SER ADYACENTES ENTRE SI EN EL ESPACIO DE DIRECCIONES VIRTUALES DEL PROCESO.
 - u ESTA RELACIONADA CON LA FORMA EN QUE SE ESCRIBEN LOS PROGRAMAS Y SE ORGANIZAN LOS DATOS.
- v “LOCALIDAD TEMPORAL”: SIGNIFICA QUE LAS LOCALIDADES DE ALMACENAMIENTO REFERENCIADAS RECIENTEMENTE TIENEN UNA ALTA PROBABILIDAD DE SER REFERENCIADAS EN UN FUTURO PROXIMO:
 - u SE APOYA EN LA UTILIZACION DE:
 - Φ FORMACION DE CICLOS (LOOPS), SUBROUTINAS, PILAS, VARIABLES USADAS PARA CONTAR Y TOTALIZAR.
- v “LOCALIDAD EN EL ESPACIO”: SIGNIFICA QUE LAS REFERENCIAS DE ALMACENAMIENTO TIENDEN A ACUMULARSE DE MANERA TAL QUE, UNA VEZ QUE SE HACE REFERENCIA A UNA LOCALIDAD, ES MUY PROBABLE QUE LAS LOCALIDADES CERCANAS SEAN TAMBIEN REFERENCIADAS:
 - u SE APOYA EN LA UTILIZACION DE:
 - Φ RECORRIDO DE ARREGLO.
 - Φ EJECUCION SECUENCIAL DE CODIGO.
 - Φ TENDENCIA DE LOS PROGRAMADORES A COLOCAR DEFINICIONES DE VARIABLES RELACIONADAS, PROXIMAS ENTRE SI.
- v UN PROGRAMA PUEDE EJECUTAR EFICIENTEMENTE MIENTRAS SU SUBCONJUNTO DE PAGINAS PREFERIDO SE ENCUENTRE EN EL

LOCALIDAD

- v EL N° DE FALLOS DE PAGINAS DE UN PROCESO DEPENDE DE LA CANTIDAD DE ALMACENAMIENTO PRIMARIO DISPONIBLE PARA SUS PAGINAS.
- v GENERALMENTE LOS PROCESOS NO MUESTRAN PATRONES DE REFERENCIAS ALEATORIOS UNIFORMEMENTE DISTRIBUIDOS POR SUS DIFERENTES PAGINAS.
- v AL REDUCIR EL N° DE MARCOS (CELDA) DE PAGINAS DISPONIBLES PARA UN PROCESO EXISTE UN INTERVALO DURANTE EL CUAL LA RAZON DE FALLOS DE PAGINAS NO SE AFECTA EXCESIVAMENTE.
- v EN DETERMINADO PUNTO, CUANDO SE REDUCE MAS EL N° DE MARCOS DE PAGINAS, EL N° DE FALLOS DE PAGINAS AUMENTA DRASTICAMENTE.
- v MIENTRAS EL SUBCONJUNTO DE PAGINAS FAVORECIDAS POR UN PROCESO PERMANEZCA EN EL ALMACENAMIENTO PRIMARIO, EL N° DE FALLOS DE PAGINAS NO AUMENTA MUCHO.
- v TAN PRONTO COMO LAS PAGINAS DEL SUBCONJUNTO FAVORECIDO SON RETIRADAS DEL ALMACENAMIENTO PRIMARIO, LA ACTIVIDAD DE PAGINACION DEL PROCESO AUMENTA EN GRAN MEDIDA AL REFERENCIAR Y TRAER DE NUEVO ESTAS PAGINAS AL ALMACENAMIENTO PRIMARIO.
- v LOS “SUBCONJUNTOS FAVORECIDOS” TAMBIEN SON LLAMADOS “CONJUNTOS DE TRABAJO” O “WORKING SETS”.

FENOMENO DE LOCALIDAD



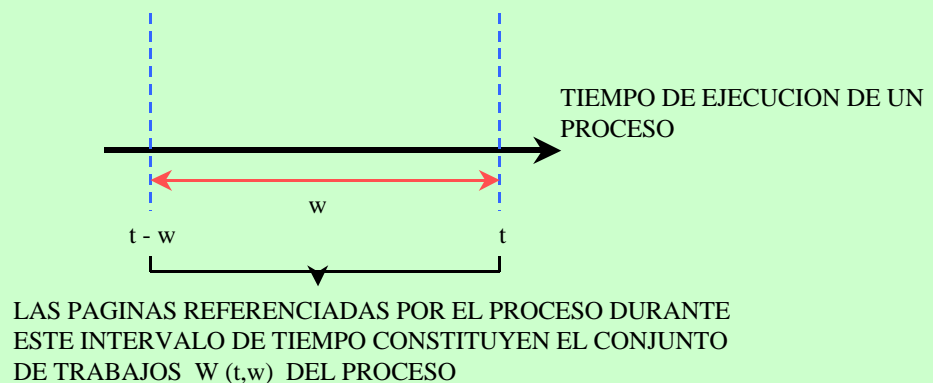
CONJUNTOS DE TRABAJO

- v DENNING DESARROLLO UN PUNTO DE VISTA DE LA ACTIVIDAD DE PAGINACION DE UN PROGRAMA LLAMADO LA “TEORIA DE CONJUNTO DE TRABAJO DEL COMPORTAMIENTO DE UN PROGRAMA”.
- v UN “CONJUNTO DE TRABAJO” ES UNA COLECCION DE PAGINAS A LAS CUALES UN PROCESO HACE ACTIVAMENTE REFERENCIA.
- v DENNING SOSTENIA QUE PARA QUE UN PROGRAMA SE EJECUTARA EFICIENTEMENTE SU CONJUNTO DE TRABAJO DEBE SER MANTENIDO EN EL ALMACENAMIENTO PRIMARIO:
 - SE EVITA LA “HIPERPAGINACION”.
- v UNA “POLITICA DE ADMINISTRACION DE ALMACENAMIENTO POR CONJUNTO DE TRABAJO” TRATA DE MANTENER EL CONJUNTO DE TRABAJO DE LOS PROGRAMAS ACTIVOS EN EL ALMACENAMIENTO PRIMARIO.
- v LA DECISION DE AÑADIR UN NUEVO PROCESO AL CONJUNTO ACTIVO DE PROCESOS (AUMENTAR EL NIVEL DE MULTIPROGRAMACION):
 - SE BASA EN SI HAY SUFICIENTE ESPACIO DISPONIBLE EN EL ALMACENAMIENTO PRIMARIO COMO PARA ACOMODAR EL CONJUNTO DE TRABAJO DEL NUEVO PROCESO.
 - SE TOMA GENERALMENTE DE FORMA HEURISTICA YA QUE ES IMPOSIBLE PARA EL SISTEMA CONOCER POR ANTICIPADO EL TAMAÑO DEL CONJUNTO DE TRABAJO DE UN PROCESO DADO.
- v EL CONJUNTO DE TRABAJO DE PAGINAS DE UN PROCESO “ $W(t,w)$ ” EN EL MOMENTO “ t ” ES EL CONJUNTO DE PAGINAS REFERIDAS POR UN PROCESO DURANTE EL INTERVALO DE TIEMPO DEL PROCESO “ $t - w$ ” A “ t ”.
- v EL “TIEMPO DEL PROCESO” ES EL TIEMPO DURANTE EL CUAL ESTE PROCESO TIENE LA CPU.
- v LA VARIABLE “ w ” SE DENOMINA “TAMAÑO DE LA VENTANA DEL CONJUNTO DE TRABAJO”:
 - LA DETERMINACION DEL TAMAÑO DE “ w ” ES MUY IMPORTANTE.
 - AL AUMENTAR EL TAMAÑO DE LA VENTANA “ w ” AUMENTA EL TAMAÑO DEL CONJUNTO DE TRABAJO.
- v “EL VERDADERO CONJUNTO DE TRABAJO DE UN PROCESO ES EL CONJUNTO DE PAGINAS QUE DEBEN ESTAR EN EL ALMACENAMIENTO PRIMARIO PARA LA EJECUCION EFICAZ DE ESTE PROCESO”.

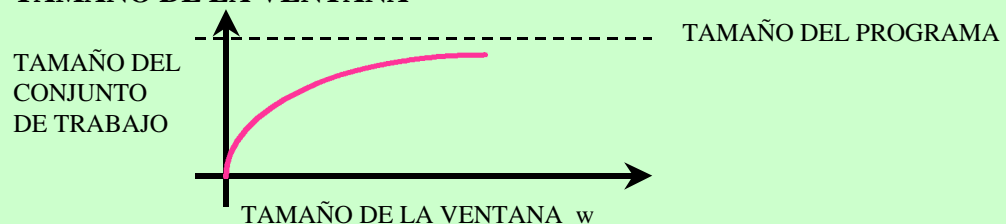
CONJUNTOS DE TRABAJO

- v LOS CONJUNTOS DE TRABAJO CAMBIAN MIENTRAS UN PROCESO ESTA EN EJECUCION:
 - v COMPLICA LA ADMINISTRACION PRECISA DEL ALMACENAMIENTO PRIMARIO EN BASE A ESTA ESTRATEGIA.
 - v “LOS CONJUNTOS DE TRABAJO SON TRANSITORIOS Y EL SIGUIENTE CONJUNTO DE TRABAJO DEL PROCESO PUEDE DIFERIR SUBSTANCIALMENTE DE SU CONJUNTO DE TRABAJO ANTERIOR”.
 - v SE DEBE EVITAR UN EXCESO DE COMPROMISO DEL ALMACENAMIENTO PRIMARIO Y LA CONSECUENTE HIPERPAGINACION.

UNA DEFINICION DEL CONJUNTO DE TRABAJO DE PAGINAS DE UN PROCESO



TAMAÑO DEL CONJUNTO DE TRABAJO COMO UNA FUNCION DEL TAMAÑO DE LA VENTANA



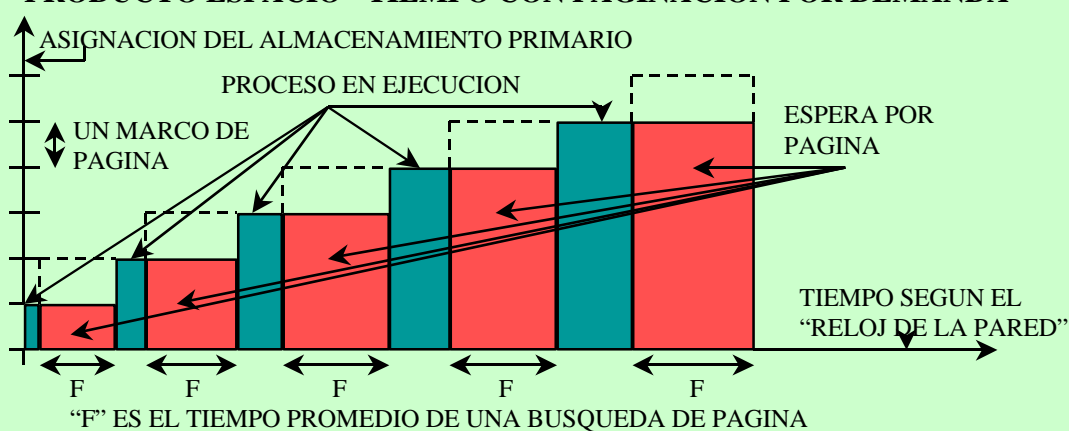
PAGINACION POR DEMANDA Y PAGINACION ANTICIPADA

- v **PAGINACION POR DEMANDA:**
- v LAS PAGINAS SON CARGADAS POR DEMANDA.
- v NO SE LLEVAN PAGINAS DEL ALMACENAMIENTO SECUNDARIO AL PRIMARIO HASTA QUE SON REFERENCIADAS EXPLICITAMENTE POR UN PROCESO EN EJECUCION.
- v LAS RAZONES DEL ATRACTIVO DE ESTA ESTRATEGIA SON:
 - u LOS RESULTADOS DE COMPUTABILIDAD, EN ESPECIAL EL “PROBLEMA DE PARADA”, INDICAN QUE EL CAMINO QUE TOMARA LA EJECUCION DE UN PROGRAMA NO SE PUEDE PREDECIR CON EXACTITUD.
 - u GARANTIZA QUE SOLO LAS PAGINAS QUE NECESITA EL PROCESO SEAN TRAIIDAS AL ALMACENAMIENTO PRINCIPAL.
 - u LA SOBRECARGA DE PROCESO PARA DECIDIR QUE PAGINA TRAER AL ALMACENAMIENTO PRINCIPAL ES MINIMA.
- v EL PRINCIPAL INCONVENIENTE ESTA EN LOS PROCESOS QUE REQUIEREN ACUMULAR SUS PAGINAS UNA POR UNA:
 - u LOS TIEMPOS DE ESPERA DE PAGINAS SON CONSIDERABLES.
 - u ES CRECIENTE LA CANTIDAD DE ALMACENAMIENTO PRIMARIO AFECTADA AL PROCESO QUE ESPERA PAGINAS:
 - Φ EL “PRODUCTO ESPACIO - TIEMPO” SE INCREMENTA.
- v EL “PRODUCTO ESPACIO - TIEMPO” INDICA LA CANTIDAD DE ALMACENAMIENTO QUE USA UN PROCESO Y LA CANTIDAD DE TIEMPO QUE LO USA.
- v “LA REDUCCION DEL PRODUCTO ESPACIO - TIEMPO DE LAS ESPERAS DE PAGINAS DE UN PROCESO ES UNA META IMPORTANTE DE LAS ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO”.
- v **PAGINACION ANTICIPADA:**
- v EL S. O. INTENTA PREDECIR LAS PAGINAS QUE UN PROCESO VA A NECESITAR Y A CONTINUACION PRECARGA ESTAS PAGINAS CUANDO HAY ESPACIO DISPONIBLE.
- v MIENTRAS EL PROCESO EJECUTA SUS PAGINAS ACTUALES, EL SISTEMA CARGA PAGINAS NUEVAS QUE ESTARAN DISPONIBLES CUANDO EL PROCESO LAS PIDA:

PAGINACION POR DEMANDA Y PAGINACION ANTICIPADA

LIBERACION DE PAGINA Y TAMAÑO DE PAGINA

PRODUCTO ESPACIO - TIEMPO CON PAGINACION POR DEMANDA



- v **LIBERACION DE PAGINA:**
- v UN PROCESO USUARIO PUEDE EMITIR UNA "LIBERACION VOLUNTARIA DE PAGINA" PARA LIBERAR EL MARCO DE PAGINA:
 - v CUANDO YA NO NECESITARA ESA PAGINA.
- v SE PUEDE ELIMINAR EL "DESPERDICIO" Y ACELERAR LA EJECUCION.
- v EL INCONVENIENTE ES QUE LA INCORPORACION DE MANDATOS DE LIBERACION DE PAGINAS DENTRO DE LOS PROGRAMAS DE USUARIOS PUEDE:
 - v SER PELIGROSO Y RETRASAR EL DESARROLLO DE APLICACIONES.
- v "LOS COMPILADORES Y S. O. DEBERIAN DETECTAR AUTOMATICAMENTE SITUACIONES DE LIBERACION DE PAGINA MUCHO ANTES DE LO QUE ES POSIBLE CON ESTRATEGIAS DE CONJUNTOS DE TRABAJO".
- v **TAMAÑO DE PAGINA:**
- v GENERALMENTE EL ALMACENAMIENTO REAL SE DIVIDE EN MARCOS O CELDAS DE PAGINA DE TAMAÑO FIJO.
- v LOS INTERROGANTES TIENEN QUE VER CON EL TAMAÑO DE LAS PAGINAS, SI TODAS LAS PAGINAS TENDRAN IGUAL TAMAÑO, SI EN CASO DE UTILIZAR PAGINAS DE DIFERENTE TAMAÑO LAS PAGINAS MAYORES DEBEN SER O NO MULTIPLOS ENTEROS DE LAS MENORES,

ETC.

LIBERACION DE PAGINA Y TAMAÑO DE PAGINA

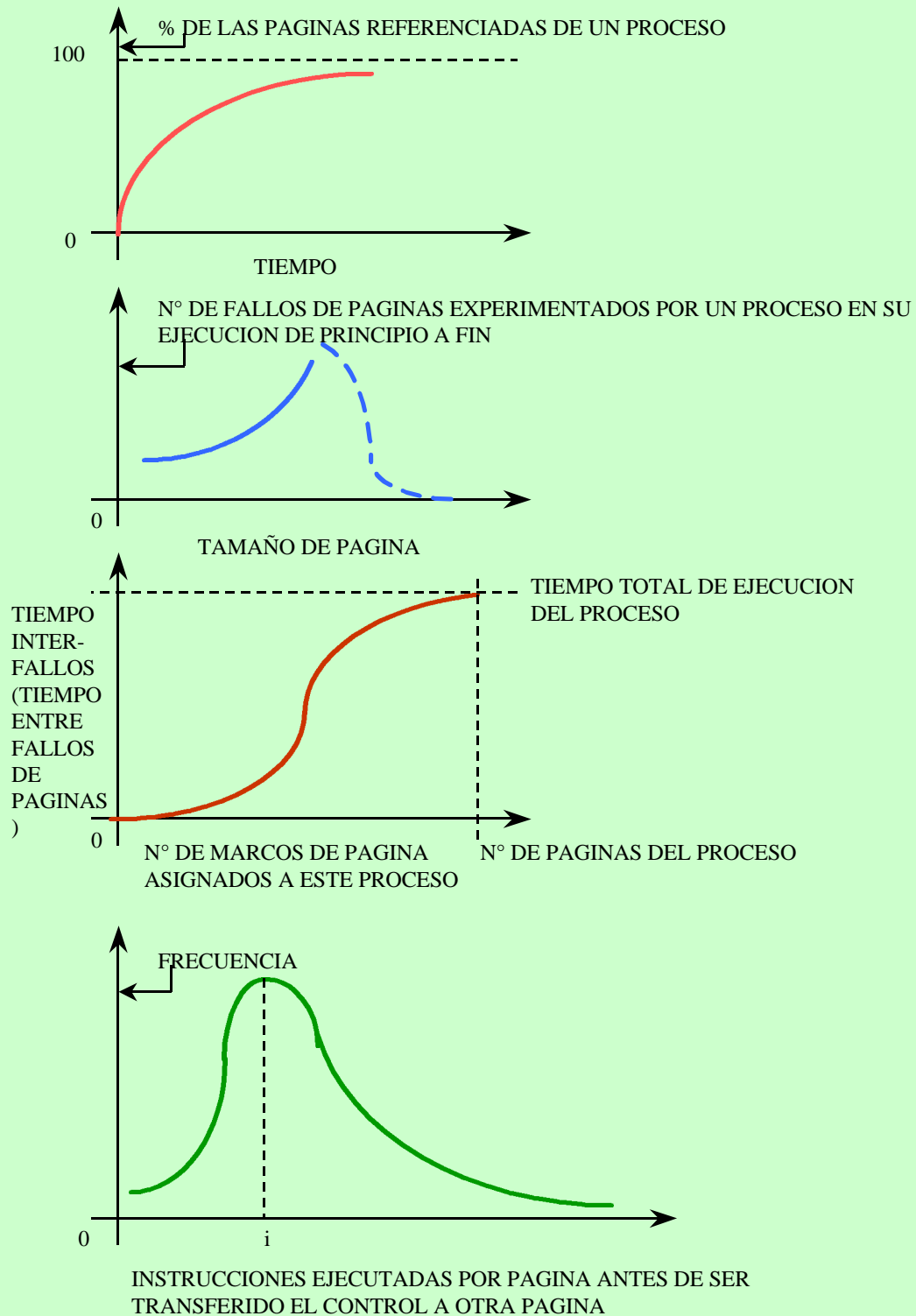
- v ALGUNAS CONSIDERACIONES PARA DETERMINAR EL TAMAÑO DE PAGINA SON LAS SIGUIENTES:
 - v CUANTO MAS PEQUEÑO SEA EL TAMAÑO DE UNA PAGINA, MAS PAGINAS Y MARCOS DE PAGINAS HABRA Y MAYORES SERAN LAS TABLAS DE PAGINAS:
 - ⊕ EL DESPERDICIO DE ALMACENAMIENTO DEBIDO AL TAMAÑO EXCESIVO DE LAS TABLAS DE PAGINA SE LLAMA “FRAGMENTACION DE TABLAS”.
 - ⊕ ESTO INDICA LA NECESIDAD DE PAGINAS MAS GRANDES.
 - v CON PAGINAS GRANDES, GRANDES CANTIDADES DE INFORMACION QUE NUNCA LLEGARIA A SER REFERENCIADA, SE PAGINARAN HACIA EL ALMACENAMIENTO PRIMARIO:
 - ⊕ ESTO INDICA LA NECESIDAD DE PAGINAS MAS PEQUEÑAS.
 - v DEBIDO A QUE LAS TRANSFERENCIAS DE E / S DEL DISCO (PAGINACION) CONSUMEN BASTANTE TIEMPO, SE DEBE MINIMIZAR LA PAGINACION QUE UN PROCESO REQUIERA:
 - ⊕ ESTO INDICA LA NECESIDAD DE PAGINAS GRANDES.
 - v LOS PROGRAMAS TIENDEN A MOSTRAR LA PROPIEDAD DE LOCALIDAD DE REFERENCIA Y ESTA LOCALIDAD TIENDE A SER PEQUEÑA:
 - ⊕ ESTO INDICA LA NECESIDAD DE PAGINAS PEQUEÑAS.
 - v LOS PROCEDIMIENTOS Y DATOS RARA VEZ COMPRENDEN UN N° ENTERO DE PAGINAS, POR LO QUE LOS SISTEMAS DE PAGINACION EXPERIMENTAN UNA “FRAGMENTACION INTERNA”:
 - ⊕ EL DESPERDICIO PROMEDIO ES DE 1 / 2 PAGINA NO USADA POR SEGMENTO (GRUPO) DE PAGINAS, QUE ESTARA EN LA ULTIMA PAGINA DEL SEGMENTO.
 - ⊕ ESTO INDICA LA NECESIDAD DE PAGINAS PEQUEÑAS.
- v LOS TAMAÑOS DE PAGINA MAS UTILIZADOS SON:
 - v 512 B, 1 KB, 2 KB, 4 KB.

COMPORTAMIENTO DE UN PROGRAMA EN LA PAGINACION

- v RESPECTO DEL PORCENTAJE DE LAS PAGINAS DE UN PROCESO TIPICO REFERENCIADAS DESDE EL MOMENTO DE INICIARSE SU EJECUCION:
 - υ UN PROCESO TIENDE A HACER REFERENCIA A UNA PARTE SIGNIFICATIVA DE SUS PAGINAS INMEDIATAMENTE DESPUES DE INICIAR SU EJECUCION.
 - υ EL PROCESO PUEDE CONCLUIR SIN HABER REFERENCIADO A ALGUNAS DE SUS PAGINAS:
 - ⊕ RUTINAS QUE ATIENDEN ERRORES QUE NO SE PRODUJERON.
- v RESPECTO DE VARIAR EL TAMAÑO DE LA PAGINA MANTENIENDO CONSTANTE LA CANTIDAD DE ALMACENAMIENTO PRIMARIO:
 - υ EL N° DE FALLOS DE PAGINAS EXPERIMENTADOS POR UN PROCESO EN EJECUCION TIENDE A AUMENTAR CON EL TAMAÑO DE LA PAGINA:
 - ⊕ SE TRAEN AL ALMACENAMIENTO PRIMARIO UN MAYOR N° DE PROCEDIMIENTOS Y DATOS QUE NO SERAN REFERENCIADOS, RESTANDO LUGAR PARA LOS QUE SI LO SERAN.
- v RESPECTO DE COMO EL PROMEDIO DE TIEMPO INTERFALLOS (TIEMPO ENTRE FALLOS DE PAGINA) VARIA AL AUMENTAR EL N° DE MARCOS DE PAGINA ASIGNADOS AL PROCESO:
 - υ CUANTO MAS MARCOS DE PAGINA TENGA UN PROCESO, MAYOR SERA EL TIEMPO ENTRE LOS FALLOS DE PAGINAS.
 - υ EL PUNTO DE INFLEXION SE DA CUANDO EL PROCESO TIENE TODO SU CONJUNTO DE TRABAJO EN EL ALMACENAMIENTO PRIMARIO.
 - υ ASIGNAR MARCOS DE PAGINA ADICIONALES MAS ALLA DEL PUNTO DE INFLEXION NO PRODUCE EFECTOS SIGNIFICATIVOS SOBRE EL TIEMPO INTERFALLOS.
- v RESPECTO DEL PORCENTAJE DE INSTRUCCIONES DE UNA PAGINA QUE SON EJECUTADAS ANTES DE TRANSFERIRSE EL CONTROL A OTRA PAGINA:
 - υ LOS VALORES EXPERIMENTALES OBTENIDOS INDICAN UN

MAXIMO DE 200 INSTRUCCIONES POR PAGINA DE 1 KB.

COMPORTAMIENTO DE UN PROGRAMA EN LA PAGINACION



SISTEMAS DE ARCHIVOS

- v INTRODUCCION
- v FUNCIONES DEL SISTEMA DE ARCHIVOS
- v EL SISTEMA DE ARCHIVOS
- v ARCHIVOS
- v DIRECTORIOS
- v IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO
- v DESCRIPTOR DE ARCHIVOS
- v SEGURIDAD
- v MECANISMOS DE PROTECCION
- v RESPALDO Y RECUPERACION

INTRODUCCION

- v TODAS LAS APLICACIONES COMPUTARIZADAS NECESITAN ALMACENAR Y RECUPERAR LA INFORMACION:
 - u SUPERANDO LAS LIMITACIONES DEL ALMACENAMIENTO REAL.
 - u TRASCENDIENDO A LA DURACION DE LOS PROCESOS QUE LAS UTILIZAN O GENERAN.
 - u INDEPENDIZANDO A LA INFORMACION DE LOS PROCESOS PERMITIENDO EL ACCESO A LA MISMA A TRAVES DE VARIOS PROCESOS.
- v LAS CONDICIONES ESENCIALES PARA EL ALMACENAMIENTO DE LA INFORMACION A LARGO PLAZO SON:
 - u DEBE SER POSIBLE ALMACENAR UNA CANTIDAD MUY GRANDE DE INFORMACION.
 - u LA INFORMACION DEBE SOBREVIVIR A LA CONCLUSION DEL PROCESO QUE LA UTILIZA.
 - u DEBE SER POSIBLE QUE VARIOS PROCESOS TENGAN ACCESO CONCURRENTENTE A LA INFORMACION.
- v LA SOLUCION ES EL ALMACENAMIENTO DE LA INFORMACION EN DISCOS Y OTROS MEDIOS EXTERNOS EN UNIDADES LLAMADAS **ARCHIVOS**:
 - u LOS ARCHIVOS DEBEN SER **PERSISTENTES**:
 - Φ NO DEBEN VERSE AFECTADOS POR LA CREACION O TERMINACION DE UN PROCESO.
 - u LOS ARCHIVOS SON UNA COLECCION DE DATOS CON NOMBRE.
 - u PUEDEN SER MANIPULADOS COMO UNA UNIDAD POR OPERACIONES COMO:
 - Φ OPEN, CLOSE, CREATE, DESTROY, COPY, RENAME, LIST.
 - u LOS ELEMENTOS DE DATOS INDIVIDUALES DENTRO DEL ARCHIVO PUEDEN SER MANIPULADOS POR OPERACIONES COMO:
 - Φ READ, WRITE, UPDATE, INSERT, DELETE.
- v EL “SISTEMA DE ARCHIVOS” ES LA PARTE DEL SISTEMA DE ADMINISTRACION DEL ALMACENAMIENTO RESPONSABLE, PRINCIPALMENTE, DE LA ADMINISTRACION DE LOS ARCHIVOS DEL ALMACENAMIENTO SECUNDARIO.
- v ES LA PARTE DEL S. O. RESPONSABLE DE PERMITIR “COMPARTIR CONTROLADAMENTE” LA INFORMACION DE LOS ARCHIVOS.

FUNCIONES DEL SISTEMA DE ARCHIVOS

- v LOS USUARIOS DEBEN PODER CREAR, MODIFICAR Y BORRAR ARCHIVOS.
- v SE DEBEN PODER COMPARTIR LOS ARCHIVOS DE UNA MANERA CUIDADOSAMENTE CONTROLADA.
- v EL MECANISMO ENCARGADO DE COMPARTIR LOS ARCHIVOS DEBE PROPORCIONAR VARIOS TIPOS DE ACCESO CONTROLADO:
 - u EJ.: “ACCESO DE LECTURA”, “ACCESO DE ESCRITURA”, “ACCESO DE EJECUCION”, VARIAS COMBINACIONES DE ESTOS, ETC.
- v SE DEBE PODER ESTRUCTURAR LOS ARCHIVOS DE LA MANERA MAS APROPIADA A CADA APLICACION.
- v LOS USUARIOS DEBEN PODER ORDENAR LA TRANSFERENCIA DE INFORMACION ENTRE ARCHIVOS.
- v SE DEBEN PROPORCIONAR POSIBILIDADES DE “RESPALDO” Y “RECUPERACION” PARA PREVENIRSE CONTRA:
 - u LA PERDIDA ACCIDENTAL DE INFORMACION.
 - u LA DESTRUCCION MALICIOSA DE INFORMACION.
- v SE DEBE PODER REFERENCIAR A LOS ARCHIVOS MEDIANTE “NOMBRES SIMBOLICOS”, BRINDANDO “INDEPENDENCIA DE DISPOSITIVOS”.
- v EN AMBIENTES SENSIBLES, EL SISTEMA DE ARCHIVOS DEBE PROPORCIONAR POSIBILIDADES DE “CIFRADO” Y “DESCIFRADO”.
- v EL SISTEMA DE ARCHIVOS DEBE BRINDAR UNA INTERFASE FAVORABLE AL USUARIO:
 - u DEBE SUMINISTRAR UNA “VISION LOGICA” DE LOS DATOS Y DE LAS FUNCIONES QUE SERAN EJECUTADAS, EN VEZ DE UNA “VISION FISICA”.
 - u EL USUARIO NO DEBE TENER QUE PREOCUPARSE POR:
 - Φ LOS DISPOSITIVOS PARTICULARES.
 - Φ DONDE SERAN ALMACENADOS LOS DATOS.
 - Φ EL FORMATO DE LOS DATOS EN LOS DISPOSITIVOS.
 - Φ LOS MEDIOS FISICOS DE LA TRANSFERENCIA DE DATOS HACIA Y DESDE LOS DISPOSITIVOS.

EL SISTEMA DE ARCHIVOS

- v UN “ARCHIVO” ES UN CONJUNTO DE REGISTROS RELACIONADOS.
- v EL “SISTEMA DE ARCHIVOS” ES UN COMPONENTE IMPORTANTE DE UN S. O. Y SUELE CONTENER:
 - u “METODOS DE ACCESO” RELACIONADOS CON LA MANERA DE ACCEDER A LOS DATOS ALMACENADOS EN ARCHIVOS.
 - u “ADMINISTRACION DE ARCHIVOS” REFERIDA A LA PROVISION DE MECANISMOS PARA QUE LOS ARCHIVOS SEAN ALMACENADOS, REFERENCIADOS, COMPARTIDOS Y ASEGURADOS.
 - u “ADMINISTRACION DEL ALMACENAMIENTO AUXILIAR” PARA LA ASIGNACION DE ESPACIO A LOS ARCHIVOS EN LOS DISPOSITIVOS DE ALMACENAMIENTO SECUNDARIO.
 - u “INTEGRIDAD DEL ARCHIVO” PARA GARANTIZAR LA INTEGRIDAD DE LA INFORMACION DEL ARCHIVO.
- v EL SISTEMA DE ARCHIVOS ESTA RELACIONADO ESPECIALMENTE CON LA ADMINISTRACION DEL ESPACIO DE ALMACENAMIENTO SECUNDARIO:
 - u FUNDAMENTALMENTE CON EL ALMACENAMIENTO DE DISCO.
- v UNA FORMA DE ORGANIZACION DE UN SISTEMA DE ARCHIVOS PUEDE SER LA SIGUIENTE:
 - u SE UTILIZA UNA “RAIZ” PARA INDICAR EN QUE PARTE DEL DISCO COMIENZA EL “DIRECTORIO RAIZ”.
 - u EL “DIRECTORIO RAIZ” APUNTA A LOS “DIRECTORIOS DE USUARIOS”.
 - u UN “DIRECTORIO DE USUARIO” CONTIENE UNA ENTRADA PARA CADA UNO DE LOS ARCHIVOS DEL USUARIO.
 - u CADA ENTRADA DE ARCHIVO APUNTA AL LUGAR DEL DISCO DONDE ESTA ALMACENADO EL ARCHIVO REFERENCIADO.
- v LOS NOMBRES DE ARCHIVOS SOLO NECESITAN SER UNICOS DENTRO DE UN DIRECTORIO DE USUARIO DADO.
- v EL NOMBRE DEL SISTEMA PARA UN ARCHIVO DADO DEBE SER UNICO PARA EL SISTEMA DE ARCHIVOS.
- v EN SISTEMAS DE ARCHIVO “JERARQUICOS” EL NOMBRE DEL SISTEMA PARA UN ARCHIVO SUELE ESTAR FORMADO COMO EL “NOMBRE DE LA

ARCHIVOS

- v SE CONSIDERARA EL PUNTO DE VISTA DEL USUARIO.
- v **NOMBRE DE LOS ARCHIVOS:**
- v LAS REGLAS EXACTAS PARA LOS NOMBRES DE ARCHIVOS VARIAN DE SISTEMA A SISTEMA.
- v ALGUNOS SISTEMAS DE ARCHIVOS DISTINGUEN ENTRE LAS LETRAS MAYUSCULAS Y MINUSCULAS, MIENTRAS QUE OTROS NO.
- v MUCHOS S. O. UTILIZAN NOMBRES DE ARCHIVO CON DOS PARTES, SEPARADAS POR UN PUNTO:
 - υ LA PARTE POSTERIOR AL PUNTO ES LA **EXTENSION DE ARCHIVO** Y GENERALMENTE INDICA ALGO RELATIVO AL ARCHIVO:
 - Φ LAS EXTENSIONES SUELEN SER MERAS CONVENCIONES.
- v **ESTRUCTURA DE UN ARCHIVO:**
- v LOS ARCHIVOS SE PUEDEN ESTRUCTURAR DE VARIAS MANERAS; LAS MAS COMUNES SON:
 - υ SECUENCIA DE BYTES, SECUENCIA DE REGISTROS Y ARBOL.
- v “SECUENCIA DE BYTES”:
 - υ EL ARCHIVO ES UNA SERIE NO ESTRUCTURADA DE BYTES.
 - υ POSEE MAXIMA FLEXIBILIDAD.
 - υ EL S. O. NO AYUDA PERO TAMPOCO ESTORBA.
- v “SECUENCIA DE REGISTROS”:
 - υ EL ARCHIVO ES UNA SECUENCIA DE REGISTROS DE LONGITUD FIJA, CADA UNO CON SU PROPIA ESTRUCTURA INTERNA.
- v “ARBOL”:
 - υ EL ARCHIVO CONSTA DE UN ARBOL DE REGISTROS, NO NECESARIAMENTE DE LA MISMA LONGITUD.
 - υ CADA REGISTRO TIENE UN CAMPO **KEY (LLAVE O CLAVE)** EN UNA POSICION FIJA DEL REGISTRO.
 - υ EL ARBOL SE ORDENA MEDIANTE EL CAMPO DE CLAVE PARA PERMITIR UNA RAPIDA BUSQUEDA DE UNA CLAVE PARTICULAR.

ARCHIVOS

v **TIPOS DE ARCHIVOS:**

v MUCHOS S. O. SOPORTAN VARIOS TIPOS DE ARCHIVOS:

- u EJ.: ARCHIVOS REGULARES, DIRECTORIOS, ARCHIVOS ESPECIALES DE CARACTERES, ARCHIVOS ESPECIALES DE BLOQUES, ETC.

v LOS **ARCHIVOS REGULARES** SON AQUELLOS QUE CONTIENEN INFORMACION DEL USUARIO.

v LOS **DIRECTORIOS** SON ARCHIVOS DE SISTEMA PARA EL MANTENIMIENTO DE UNA ESTRUCTURA DEL SISTEMA DE ARCHIVOS.

v LOS **ARCHIVOS ESPECIALES DE CARACTERES:**

- u TIENEN RELACION CON LA E / S.
- u SE UTILIZAN PARA MODELAR DISPOSITIVOS SERIALES DE E / S (TERMINALES, IMPRESORAS, REDES, ETC.).

v LOS **ARCHIVOS ESPECIALES DE BLOQUES** SE UTILIZAN PARA MODELAR DISCOS.

v **ACCESO A UN ARCHIVO:**

v LOS TIPOS DE ACCESO MAS CONOCIDOS SON:

- u **ACCESO SECUENCIAL:** EL PROCESO LEE EN ORDEN TODOS LOS REGISTROS DEL ARCHIVO COMENZANDO POR EL PRINCIPIO, SIN PODER:

- Φ SALTAR REGISTROS.
- Φ LEER EN OTRO ORDEN.

- u **ACCESO ALEATORIO:** EL PROCESO PUEDE LEER LOS REGISTROS EN CUALQUIER ORDEN UTILIZANDO DOS METODOS PARA DETERMINAR EL PUNTO DE INICIO DE LA LECTURA:

- Φ CADA OPERACION DE LECTURA (READ) DA LA POSICION EN EL ARCHIVO CON LA CUAL INICIAR.
- Φ UNA OPERACION ESPECIAL (SEEK) ESTABLECE LA POSICION DE TRABAJO PUDIENDO LUEGO LEERSE EL ARCHIVO SECUENCIALMENTE.

ARCHIVOS

v **ATRIBUTOS DE ARCHIVO:**

v CADA ARCHIVO TIENE:

- v SU NOMBRE Y DATOS.
- v ELEMENTOS ADICIONALES LLAMADOS **ATRIBUTOS**, QUE VARIAN CONSIDERABLEMENTE DE SISTEMA A SISTEMA.

v ALGUNOS DE LOS POSIBLES **ATRIBUTOS** DE ARCHIVO SON:

- v “PROTECCION”: QUIEN DEBE TENER ACCESO Y DE QUE FORMA.
- v “CONTRASEÑA”: CONTRASEÑA NECESARIA PARA ACCEDER AL ARCHIVO.
- v “CREADOR”: IDENTIFICADOR DE LA PERSONA QUE CREO EL ARCHIVO.
- v “PROPIETARIO”: PROPIETARIO ACTUAL.
- v “BANDERA EXCLUSIVO-PARA-LECTURA”: 0 LECTURA / ESCRITURA; 1 PARA LECTURA EXCLUSIVAMENTE.
- v “BANDERA DE OCULTAMIENTO”: 0 NORMAL, 1 PARA NO EXHIBIRSE EN LISTAS.
- v “BANDERA DE SISTEMA”: 0 ARCHIVO NORMAL, 1 ARCHIVO DE SISTEMA.
- v “BANDERA DE BIBLIOTECA”: 0 YA SE HA RESPALDADO, 1 NECESITA RESPALDO.
- v “BANDERA ASCII / BINARIO”: 0 ARCHIVO EN ASCII, 1 ARCHIVO EN BINARIO.
- v “BANDERA DE ACCESO ALEATORIO”: 0 SOLO ACCESO SECUENCIAL, 1 ACCESO ALEATORIO.
- v “BANDERA TEMPORAL”: 0 NORMAL, 1 ELIMINAR AL SALIR DEL PROCESO.
- v “BANDERAS DE CERRADURA”: 0 NO BLOQUEADO, DISTINTO DE 0 BLOQUEADO.
- v “LONGITUD DEL REGISTRO”: N° DE BYTES EN UN REGISTRO.
- v “POSICION DE LA LLAVE”: AJUSTE DE LA LLAVE DENTRO DE CADA REGISTRO.
- v “LONGITUD DE LA LLAVE”: N° DE BYTES EN EL CAMPO LLAVE.

ARCHIVOS

- υ “TIEMPO DE CREACION”: FECHA Y HORA DE CREACION DEL ARCHIVO.
- υ “TIEMPO DEL ULTIMO ACCESO”: FECHA Y HORA DEL ULTIMO ACCESO AL ARCHIVO.
- υ “TIEMPO DE LA ULTIMA MODIFICACION”: FECHA Y HORA DE LA ULTIMA MODIFICACION AL ARCHIVO.
- υ “TAMAÑO ACTUAL”: N° DE BYTES EN EL ARCHIVO.
- υ “TAMAÑO MAXIMO”: TAMAÑO MAXIMO AL QUE PUEDE CRECER EL ARCHIVO.
- υ **OPERACIONES CON ARCHIVOS:**
- υ LAS LLAMADAS MAS COMUNES AL SISTEMA RELACIONADAS CON LOS ARCHIVOS SON:
 - υ **CREATE (CREAR):** EL ARCHIVO SE CREA SIN DATOS.
 - υ **DELETE (ELIMINAR):** SI EL ARCHIVO YA NO ES NECESARIO DEBE ELIMINARSE PARA LIBERAR ESPACIO EN DISCO. CIERTOS S. O. ELIMINAN AUTOMATICAMENTE UN ARCHIVO NO UTILIZADO DURANTE “N” DIAS.
 - υ **OPEN (ABRIR):** ANTES DE UTILIZAR UN ARCHIVO, UN PROCESO DEBE ABRIRLO. LA FINALIDAD ES PERMITIR QUE EL SISTEMA TRASLADA LOS ATRIBUTOS Y LA LISTA DE DIRECCIONES EN DISCO A LA MEMORIA PRINCIPAL PARA UN RAPIDO ACCESO EN LLAMADAS POSTERIORES.
 - υ **CLOSE (CERRAR):** CUANDO CONCLUYEN LOS ACCESOS, LOS ATRIBUTOS Y DIRECCIONES DEL DISCO YA NO SON NECESARIOS, POR LO QUE EL ARCHIVO DEBE CERRARSE Y LIBERAR LA TABLA DE ESPACIO INTERNO.
 - υ **READ (LEER):** LOS DATOS SE LEEN DEL ARCHIVO. QUIEN HACE LA LLAMADA DEBE ESPECIFICAR LA CANTIDAD DE DATOS NECESARIOS Y PROPORCIONAR UN BUFFER PARA COLOCARLOS.
 - υ **WRITE (ESCRIBIR):** LOS DATOS SE ESCRIBEN EN EL ARCHIVO, EN LA POSICION ACTUAL. EL TAMAÑO DEL ARCHIVO PUEDE AUMENTAR (AGREGADO DE REGISTROS) O NO (ACTUALIZACION DE REGISTROS).

ARCHIVOS

- ▮ **APPEND (AÑADIR):** ES UNA FORMA RESTRINGIDA DE “WRITE”. SOLO PUEDE AÑADIR DATOS AL FINAL DEL ARCHIVO.
- ▮ **SEEK (BUSCAR):** ESPECIFICA EL PUNTO DONDE POSICIONARSE. CAMBIA LA POSICION DEL APUNTADOR A LA POSICION ACTIVA EN CIERTO LUGAR DEL ARCHIVO.
- ▮ **GET ATTRIBUTES (OBTENER ATRIBUTOS):** PERMITE A LOS PROCESOS OBTENER LOS ATRIBUTOS DEL ARCHIVO.
- ▮ **SET ATTRIBUTES (ESTABLECER ATRIBUTOS):** ALGUNOS ATRIBUTOS PUEDEN SER DETERMINADOS POR EL USUARIO Y MODIFICADOS LUEGO DE LA CREACION DEL ARCHIVO. LA INFORMACION RELATIVA AL MODO DE PROTECCION Y LA MAYORIA DE LAS BANDERAS SON UN EJEMPLO OBVIO.
- ▮ **RENAME (CAMBIAR DE NOMBRE):** PERMITE MODIFICAR EL NOMBRE DE UN ARCHIVO YA EXISTENTE.
- ▮ **ARCHIVOS MAPEADOS A MEMORIA:**
- ▮ ALGUNOS S. O. PERMITEN ASOCIAR LOS ARCHIVOS CON UN ESPACIO DE DIRECCIONES DE UN PROCESO EN EJECUCION.
- ▮ SE UTILIZAN LAS LLAMADAS AL SISTEMA “MAP” Y “UNMAP”.
- ▮ “MAP” UTILIZA UN NOMBRE DE ARCHIVO Y UNA DIRECCION VIRTUAL Y HACE QUE EL S. O. ASOCIE AL ARCHIVO CON LA DIRECCION VIRTUAL EN EL ESPACIO DE DIRECCIONES:
 - ▮ LAS LECTURAS O ESCRITURAS DE LAS AREAS DE MEMORIA ASOCIADAS AL ARCHIVO SE EFECTUAN TAMBIEN SOBRE EL ARCHIVO MAPEADO.
- ▮ “UNMAP” ELIMINA LOS ARCHIVOS DEL ESPACIO DE DIRECCIONES Y CONCLUYE LA OPERACION DE ASOCIACION.
- ▮ EL MAPEO DE ARCHIVOS ELIMINA LA NECESIDAD DE PROGRAMAR LA E / S DIRECTAMENTE, FACILITANDO LA PROGRAMACION.
- ▮ LOS PRINCIPALES PROBLEMAS RELACIONADOS SON:
 - ▮ IMPOSIBILIDAD DE CONOCER A PRIORI LA LONGITUD DEL ARCHIVO DE SALIDA, EL QUE PODRIA SUPERAR A LA MEMORIA.
 - ▮ DIFICULTAD PARA COMPARTIR LOS ARCHIVOS MAPEADOS EVITANDO INCONSISTENCIAS, YA QUE LAS MODIFICACIONES HECHAS EN LAS PAGINAS NO SE VERAN REFLEJADAS EN EL DISCO HASTA QUE DICHAS PAGINAS SEAN ELIMINADAS DE LA MEMORIA.

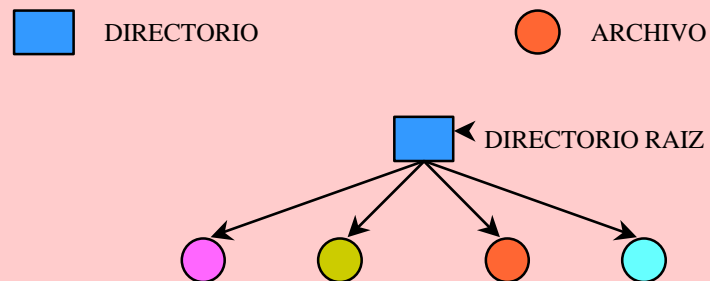
DIRECTORIOS

- v GENERALMENTE SON UTILIZADOS POR LOS S. O. PARA LLEVAR UN REGISTRO DE LOS ARCHIVOS.
- v EN MUCHOS SISTEMAS SON A SU VEZ TAMBIEN ARCHIVOS.
- v **SISTEMAS JERARQUICOS DE DIRECTORIOS:**
- v EL DIRECTORIO CONTIENE UN CONJUNTO DE DATOS POR CADA ARCHIVO REFERENCIADO.
- v UNA POSIBILIDAD ES QUE EL DIRECTORIO CONTENGA POR CADA ARCHIVO REFERENCIADO EL NOMBRE, SUS ATRIBUTOS Y LAS DIRECCIONES EN DISCO DONDE SE ALMACENAN LOS DATOS.
- v OTRA POSIBILIDAD ES QUE CADA ENTRADA DEL DIRECTORIO CONTENGA:
 - u EL NOMBRE DEL ARCHIVO.
 - u UN APUNTADOR A OTRA ESTRUCTURA DE DATOS DONDE SE ENCUENTRAN LOS ATRIBUTOS Y LAS DIRECCIONES EN DISCO.
- v AL ABRIR UN ARCHIVO EL S. O.:
 - u BUSCA EN SU DIRECTORIO EL NOMBRE DEL ARCHIVO.
 - u EXTRAE LOS ATRIBUTOS Y DIRECCIONES EN DISCO.
 - u GRABA ESTA INFORMACION EN UNA TABLA DE MEMORIA REAL.
 - u TODAS LAS REFERENCIAS SUBSECUENTES AL ARCHIVO UTILIZAN LA INFORMACION DE LA MEMORIA PRINCIPAL.
- v EL N° Y ORGANIZACION DE DIRECTORIOS VARIA DE SISTEMA EN SISTEMA:
 - u DIRECTORIO UNICO: EL SISTEMA TIENE UN DOLO DIRECTORIO CON TODOS LOS ARCHIVOS DE TODOS LOS USUARIOS.
 - u UN DIRECTORIO POR USUARIO: EL SISTEMA HABILITA UN SOLO DIRECTORIO POR CADA USUARIO.
 - u UN ARBOL DE DIRECTORIOS POR USUARIO: EL SISTEMA PERMITE QUE CADA USUARIO TENGA TANTOS DIRECTORIOS COMO NECESITE, RESPETANDO UNA JERARQUIA GENERAL.

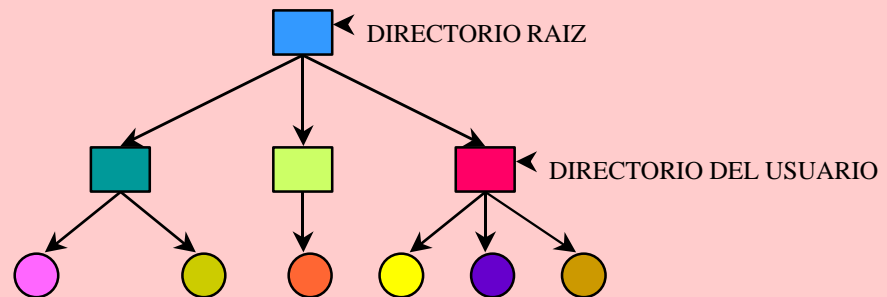
DIRECTORIOS

TRES DISEÑOS DE SISTEMAS DE ARCHIVOS

UN SOLO DIRECTORIO COMPARTIDO POR TODOS LOS USUARIOS

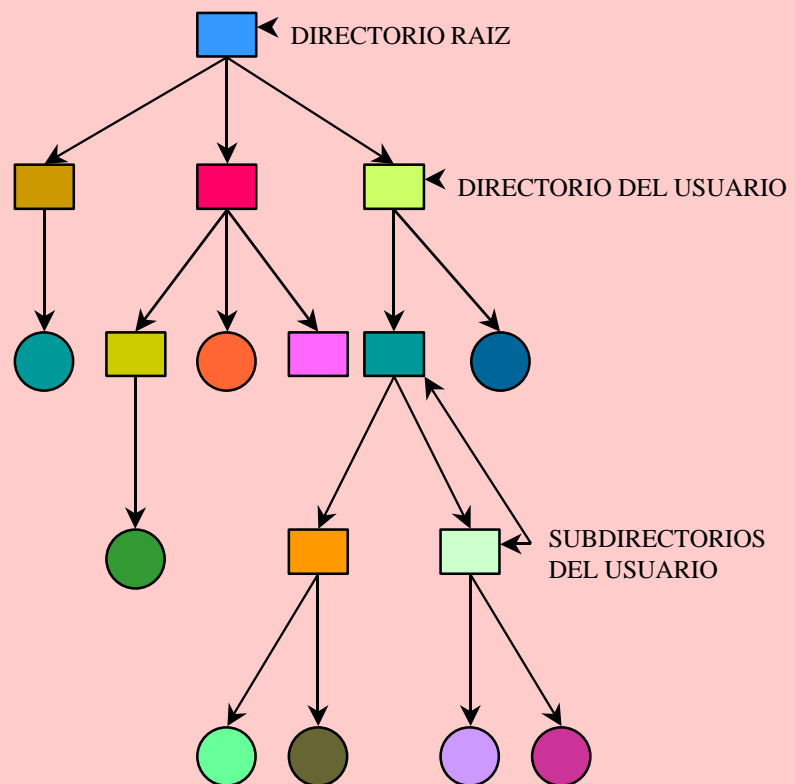


UN DIRECTORIO POR USUARIO



DIRECTORIOS

UN ARBOL ARBITRARIO POR USUARIO



DIRECTORIOS

- v **NOMBRE DE LAS RUTAS DE ACCESO:**
- v CUANDO EL SISTEMA DE ARCHIVOS ESTA ORGANIZADO COMO UN ARBOL DE DIRECTORIOS SE NECESITA UNA FORMA DE DETERMINAR LOS NOMBRES DE LOS ARCHIVOS.
- v LOS PRINCIPALES METODOS PARA NOMBRES DE LOS ARCHIVOS SON:
 - u **RUTA DE ACCESO ABSOLUTA:**
 - Φ CADA ARCHIVO TIENE UNA RUTA DE ACCESO ABSOLUTA.
 - Φ CONSTA DE LA RUTA DE ACCESO DESDE EL DIRECTORIO RAIZ HASTA EL ARCHIVO.
 - Φ LOS COMPONENTES DE LA RUTA DE ACCESO SE SEPARAN MEDIANTE ALGUN CARACTER LLAMADO “SEPARADOR”.
 - u **RUTA DE ACCESO RELATIVA:**
 - Φ SE UTILIZA JUNTO CON EL CONCEPTO DE **DIRECTORIO DE TRABAJO O DIRECTORIO ACTIVO**.
 - Φ TODOS LOS NOMBRES QUE NO COMIENCEN EN EL DIRECTORIO RAIZ SE TOMAN EN RELACION CON EL DIRECTORIO DE TRABAJO.
 - Φ EL NOMBRE ABSOLUTO DE LA RUTA DE ACCESO SIEMPRE FUNCIONA, SIN IMPORTAR CUAL SEA EL DIRECTORIO DE TRABAJO.
- v **OPERACIONES CON DIRECTORIOS:**
- v LAS LLAMADAS AL SISTEMA PERMITIDAS PARA EL MANEJO DE LOS DIRECTORIOS TIENEN VARIACION DE SISTEMA A SISTEMA.
- v LAS MAS COMUNES SON LAS SIGUIENTES:
 - u **CREATE (CREAR):** SE CREA UN DIRECTORIO VACIO.
 - u **DELETE (ELIMINAR):** SE ELIMINA UN DIRECTORIO, QUE DEBE ESTAR VACIO.
 - u **OPENDIR (ABRIR DIRECTORIO):** SE PUEDEN LEER LOS DIRECTORIOS:
 - Φ ANTES DE PODER LEER UN DIRECTORIO, ESTE DEBE SER ABIERTO.

DIRECTORIOS

- υ **CLOSEDIR (CERRAR DIRECTORIO):** CUANDO SE HA LEIDO UN DIRECTORIO, ESTE DEBE SER CERRADO PARA LIBERAR EL ESPACIO CORRESPONDIENTE DE LA TABLA INTERNA.
- υ **READDIR (LEER DIRECTORIO):** REGRESA LA SIGUIENTE ENTRADA EN UN DIRECTORIO ABIERTO, SIN IMPORTAR EL TIPO DE ESTRUCTURA DE DIRECTORIOS QUE SE UTILICE.
- υ **RENAME (CAMBIAR DE NOMBRE):** CAMBIA EL NOMBRE DE UN DIRECTORIO DE MANERA SIMILAR AL CAMBIO PARA ARCHIVOS.
- υ **LINK (LIGAR):** ES UNA TECNICA QUE PERMITE QUE UN ARCHIVO APAREZCA EN MAS DE UN DIRECTORIO:
 - Φ ESPECIFICA UN ARCHIVO EXISTENTE Y EL NOMBRE DE UNA RUTA DE ACCESO.
 - Φ CREA UN ENLACE DEL ARCHIVO YA EXISTENTE CON EL NOMBRE ESPECIFICADO EN LA RUTA DE ACCESO.
- υ **UNLINK (DESLIGAR):** SE ELIMINA UNA ENTRADA DEL DIRECTORIO:
 - Φ SI EL ARCHIVO QUE SE DESEA DESLIGAR APARECE SOLO EN UN DIRECTORIO (EL CASO NORMAL):
 - SE ELIMINA DEL SISTEMA DE ARCHIVOS.
 - Φ SI EL ARCHIVO QUE SE DESEA DESLIGAR ESTA PRESENTE EN VARIOS DIRECTORIOS:
 - SOLO SE ELIMINA LA RUTA DE ACCESO ESPECIFICADA.
 - LAS DEMAS RUTAS PERMANECEN.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v SE CONSIDERAN ASPECTOS TALES COMO:
 - u LA FORMA DE ALMACENAMIENTO DE ARCHIVOS Y DIRECTORIOS.
 - u LA ADMINISTRACION DEL ESPACIO EN DISCO.
 - u LA FORMA DE HACERLO DE MANERA EFICIENTE Y CONFIABLE.
- v SE DEBEN TENER PRESENTES PROBLEMAS TALES COMO LA “FRAGMENTACION” CRECIENTE DEL ESPACIO EN DISCO:
 - u OCASIONA PROBLEMAS DE PERFORMANCE AL HACER QUE LOS ARCHIVOS SE DESPERDIGUEN A TRAVES DE BLOQUES MUY DISPERSOS.
 - u UNA TECNICA PARA ALIVIAR EL PROBLEMA DE LA “FRAGMENTACION” CONSISTE EN REALIZAR PERIODICAMENTE:
 - Φ “CONDENSACION”:
 - SE PUEDEN “REORGANIZAR” LOS ARCHIVOS EXPRESAMENTE O AUTOMATICAMENTE SEGUN ALGUN CRITERIO PREDEFINIDO.
 - Φ “RECOLECCION DE BASURA O RESIDUOS”:
 - SE PUEDE HACER FUERA DE LINEA O EN LINEA, CON EL SISTEMA ACTIVO, SEGUN LA IMPLEMENTACION.
- v **IMPLANTACION DE ARCHIVOS:**
- v EL ASPECTO CLAVE DE LA IMPLANTACION DEL ALMACENAMIENTO DE ARCHIVOS ES EL REGISTRO DE LOS BLOQUES ASOCIADOS A CADA ARCHIVO.
- v ALGUNOS DE LOS METODOS UTILIZADOS SON LOS SIGUIENTES:
- v ASIGNACION CONTIGUA O ADYACENTE:
 - u LOS ARCHIVOS SON ASIGNADOS A AREAS CONTIGUAS DE ALMACENAMIENTO SECUNDARIO.
 - u LAS PRINCIPALES VENTAJAS SON:
 - Φ FACILIDAD DE IMPLANTACION, YA QUE SOLO SE PRECISA EL N° DEL BLOQUE DE INICIO PARA LOCALIZAR UN ARCHIVO.
 - Φ RENDIMIENTO EXCELENTE RESPECTO DE LA E / S.
 - u LOS PRINCIPALES DEFECTOS SON:
 - Φ SE DEBE CONOCER EL TAMAÑO MAXIMO DEL ARCHIVO AL CREARLO.
 - Φ PRODUCE UNA GRAN FRAGMENTACION DE LOS DISCOS.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

v ASIGNACION NO CONTIGUA:

υ SON ESQUEMAS DE ALMACENAMIENTO MAS DINAMICOS.

υ ASIGNACION ENCADENADA ORIENTADA HACIA EL SECTOR:

- ⊕ EL DISCO SE CONSIDERA COMPUESTO DE SECTORES INDIVIDUALES.
- ⊕ LOS ARCHIVOS CONSTAN DE VARIOS SECTORES QUE PUEDEN ESTAR DISPERSOS POR TODO EL DISCO.
- ⊕ LOS SECTORES QUE PERTENECEN A UN ARCHIVO COMUN CONTIENEN APUNTADES DE UNO A OTRO FORMANDO UNA “LISTA ENCADENADA”.
- ⊕ UNA “LISTA DE ESPACIO LIBRE” CONTIENE ENTRADAS PARA TODOS LOS SECTORES LIBRES DEL DISCO.
- ⊕ LAS AMPLIACIONES O REDUCCIONES EN EL TAMAÑO DE LOS ARCHIVOS SE RESUELVEN ACTUALIZANDO LA “LISTA DE ESPACIO LIBRE”:
 - NO HAY NECESIDAD DE CONDENSACION.
- ⊕ LAS PRINCIPALES DESVENTAJAS SON:
 - DEBIDO A LA POSIBLE DISPERSION EN EL DISCO, LA RECUPERACION DE REGISTROS LOGICAMENTE CONTIGUOS PUEDE SIGNIFICAR LARGAS BUSQUEDAS.
 - EL MANTENIMIENTO DE LA ESTRUCTURA DE “LISTAS ENCADENADAS” SIGNIFICA UNA SOBRECARGA EN TIEMPO DE EJECUCION.
 - LOS APUNTADES DE LA ESTRUCTURA DE LISTA CONSUMEN ESPACIO EN DISCO.

υ ASIGNACION POR BLOQUES:

- ⊕ ES MAS EFICIENTE Y REDUCE LA SOBRECARGA EN EJECUCION.
- ⊕ ES UNA MEZCLA DE LOS METODOS DE ASIGNACION CONTIGUA Y NO CONTIGUA.
- ⊕ SE ASIGNAN BLOQUES DE SECTORES CONTIGUOS EN VEZ DE SECTORES INDIVIDUALES.
- ⊕ EL SISTEMA TRATA DE ASIGNAR NUEVOS BLOQUES A UN ARCHIVO ELIGIENDO BLOQUES LIBRES LO MAS PROXIMOS POSIBLE A LOS BLOQUES DEL ARCHIVO EXISTENTES.

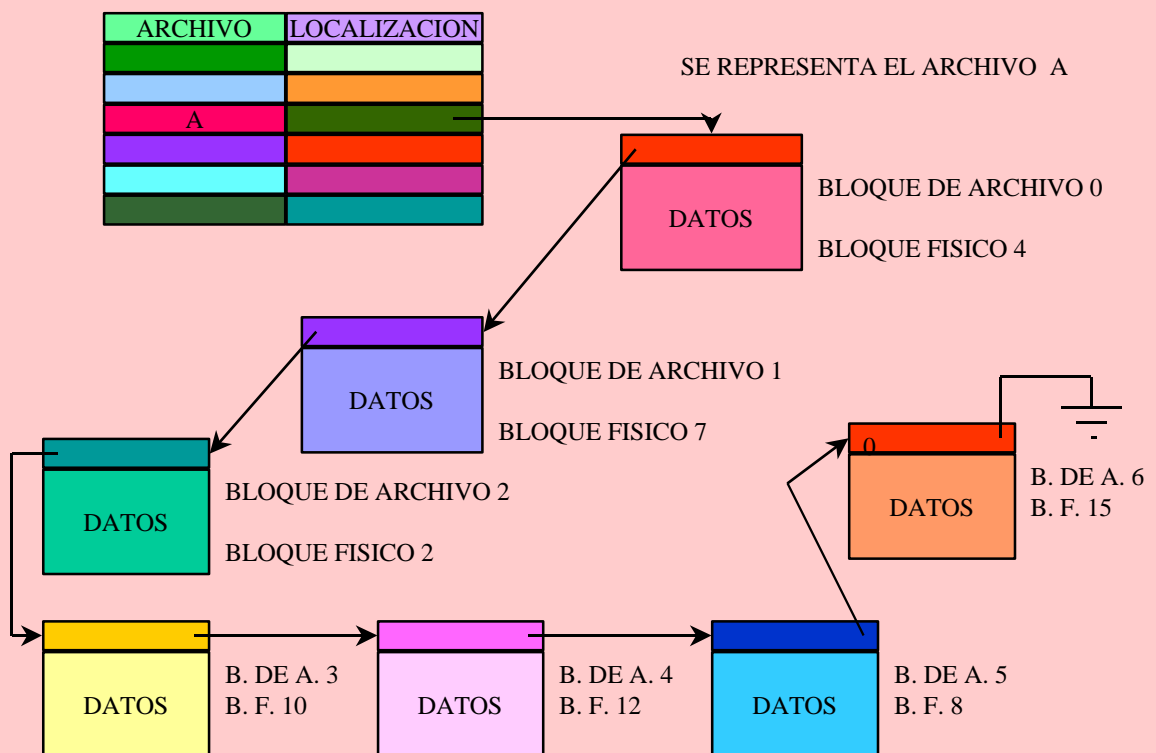
IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- υ LAS FORMAS MAS COMUNES DE IMPLEMENTAR LA ASIGNACION POR BLOQUES SON:
 - Φ ENCADENAMIENTO DE BLOQUES.
 - Φ ENCADENAMIENTO DE BLOQUES DE INDICE.
 - Φ TRANSFORMACION DE ARCHIVOS ORIENTADA HACIA BLOQUES.
- υ ENCADENAMIENTO DE BLOQUES O LISTA LIGADA:
- υ LAS ENTRADAS EN EL DIRECTORIO DE USUARIOS APUNTAN AL PRIMER BLOQUE DE CADA ARCHIVO.
- υ CADA UNO DE LOS BLOQUES DE LONGITUD FIJA QUE FORMAN UN ARCHIVO CONTIENE DOS PARTES:
 - Φ UN BLOQUE DE DATOS.
 - Φ UN APUNTADOR AL BLOQUE SIGUIENTE.
- υ CADA BLOQUE CONTIENE VARIOS SECTORES.
- υ FRECUENTEMENTE EL TAMAÑO DE UN BLOQUE SE CORRESPONDE CON EL DE UNA PISTA COMPLETA DEL DISCO.
- υ LOCALIZAR UN REGISTRO DETERMINADO REQUIERE:
 - Φ BUSCAR EN LA CADENA DE BLOQUES HASTA ENCONTRAR EL BLOQUE APROPIADO.
 - Φ BUSCAR EN EL BLOQUE HASTA ENCONTRAR EL REGISTRO.
- υ EL EXAMEN DE LA CADENA DESDE EL PRINCIPIO PUEDE SER LENTO:
 - Φ DEBE REALIZARSE DE BLOQUE EN BLOQUE, Y PUEDEN ESTAR DISPERSOS POR TODO EL DISCO.
- υ LA INSERCIÓN Y EL RETIRO SON INMEDIATOS:
 - Φ SE DEBEN MODIFICAR LOS APUNTADES DEL BLOQUE PRECEDENTE.
- υ SE PUEDEN USAR “LISTAS DE ENCADENAMIENTO DOBLE”:
 - Φ HACIA ADELANTE Y HACIA ATRAS.
 - Φ SE FACILITA LA BUSQUEDA.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

ENCADENAMIENTO DE BLOQUES O LISTA LIGADA DE BLOQUES

DIRECTORIO DE USUARIOS



ENCADENAMIENTO DE BLOQUES DE INDICES:

LOS APUNTADES SON COLOCADOS EN VARIOS BLOQUES DE INDICES SEPARADOS:

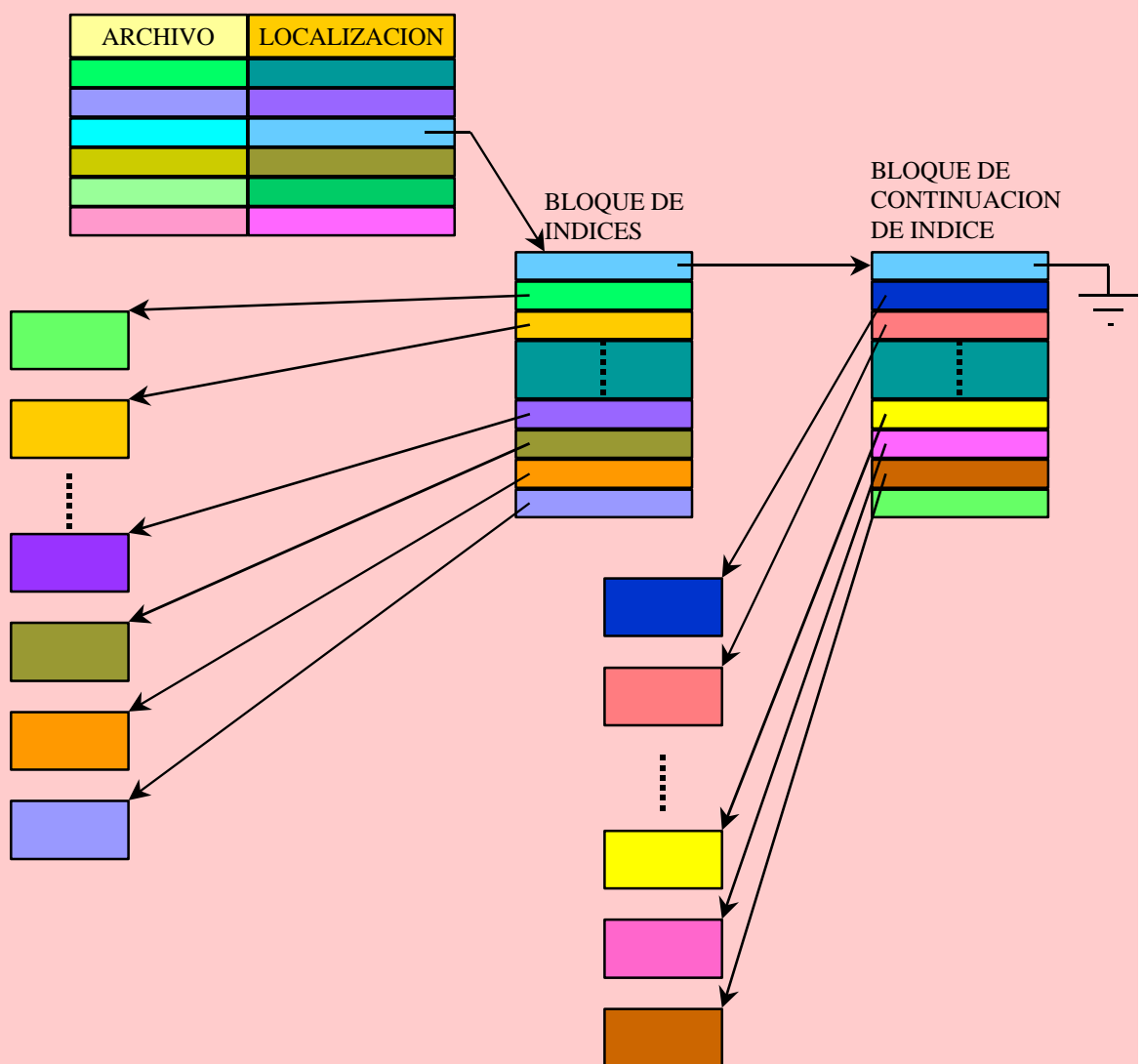
- ❖ CADA BLOQUE DE INDICES CONTIENE UN N° FIJO DE ELEMENTOS.
- ❖ CADA ENTRADA CONTIENE:
 - UN IDENTIFICADOR DE REGISTROS.
 - UN APUNTADES A ESE REGISTRO.
- ❖ SI ES NECESARIO UTILIZAR MAS DE UN BLOQUE DE INDICES PARA DESCRIBIR UN ARCHIVO:
 - SE ENCADENA UNA SERIE DE BLOQUES DE INDICES.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- υ LA GRAN VENTAJA ES QUE LA BUSQUEDA PUEDE REALIZARSE EN LOS PROPIOS BLOQUES DE INDICES.
- υ LOS BLOQUES DE INDICES PUEDEN MANTENERSE JUNTOS EN EL ALMACENAMIENTO SECUNDARIO PARA ACORTAR LA BUSQUEDA:
 - ϕ PARA MEJOR PERFORMANCE PODRIAN MANTENERSE EN EL ALMACENAMIENTO PRIMARIO.
- υ LA PRINCIPAL DESVENTAJA ES QUE LAS INSERCCIONES PUEDEN REQUERIR LA RECONSTRUCCION COMPLETA DE LOS BLOQUES DE INDICES:
 - ϕ UNA POSIBILIDAD ES DEJAR VACIA UNA PARTE DE LOS BLOQUES DE INDICES PARA:
 - FACILITAR INSERCCIONES FUTURAS.
 - RETARDAR LAS RECONSTRUCCIONES.
- υ ES SUFICIENTE QUE EL DATO DEL DIRECTORIO CONTENGA EL N° DE BLOQUE INICIAL PARA LOCALIZAR TODOS LOS BLOQUES RESTANTES:
 - ϕ SIN IMPORTAR EL TAMAÑO DEL ARCHIVO.
- υ TRANSFORMACION DE ARCHIVOS ORIENTADA HACIA BLOQUES:
- υ SE UTILIZAN NUMEROS DE BLOQUES EN VEZ DE APUNTADES.
- υ LOS N° DE BLOQUES SE CONVIERTEN FACILMENTE A DIRECCIONES DE BLOQUES GRACIAS A LA GEOMETRIA DEL DISCO.
- υ SE CONSERVA UN MAPA DEL ARCHIVO, CONTENIENDO UNA ENTRADA PARA CADA BLOQUE DEL DISCO.
- υ LAS ENTRADAS EN EL DIRECTORIO DEL USUARIO APUNTAN A LA PRIMERA ENTRADA AL MAPA DEL ARCHIVO PARA CADA ARCHIVO.
- υ CADA ENTRADA AL MAPA DEL ARCHIVO CONTIENE EL N° DEL BLOQUE SIGUIENTE DE ESE ARCHIVO.
- υ LA ENTRADA AL MAPA DEL ARCHIVO CORRESPONDIENTE A LA ULTIMA ENTRADA DE UN ARCHIVO DETERMINADO SE AJUSTA A ALGUN VALOR “CENTINELA” (“NIL”) PARA INDICAR QUE SE ALCANZO EL ULTIMO BLOQUE DE UN ARCHIVO.
- υ EL SISTEMA PUEDE MANTENER UNA LISTA DE BLOQUES LIBRES.
- υ LA PRINCIPAL VENTAJA ES QUE LAS CERCANIAS FISICAS DEL DISCO SE REFLEJAN EN EL MAPA DEL ARCHIVO.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

ENCADENAMIENTO DE BLOQUES DE INDICES



IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

TRANSFORMACION DE ARCHIVOS ORIENTADA HACIA BLOQUES

DIRECTORIO DE USUARIOS

ARCHIVO	LOCALIZACION
A	8
B	6
C	2

MAPA DEL ARCHIVO

0	22
1	NIL
2	5
3	26
4	9
5	20
6	10
7	LIBRE
8	17
9	1
10	14
11	LIBRE
12	3
13	4
14	0
15	LIBRE
16	LIBRE
17	12
18	13
19	NIL
20	23
21	LIBRE
22	18
23	19
24	LIBRE
25	LIBRE
26	NIL
27	LIBRE

BLOQUES FISICOS EN EL ALMACENAMIENTO SECUNDARIO

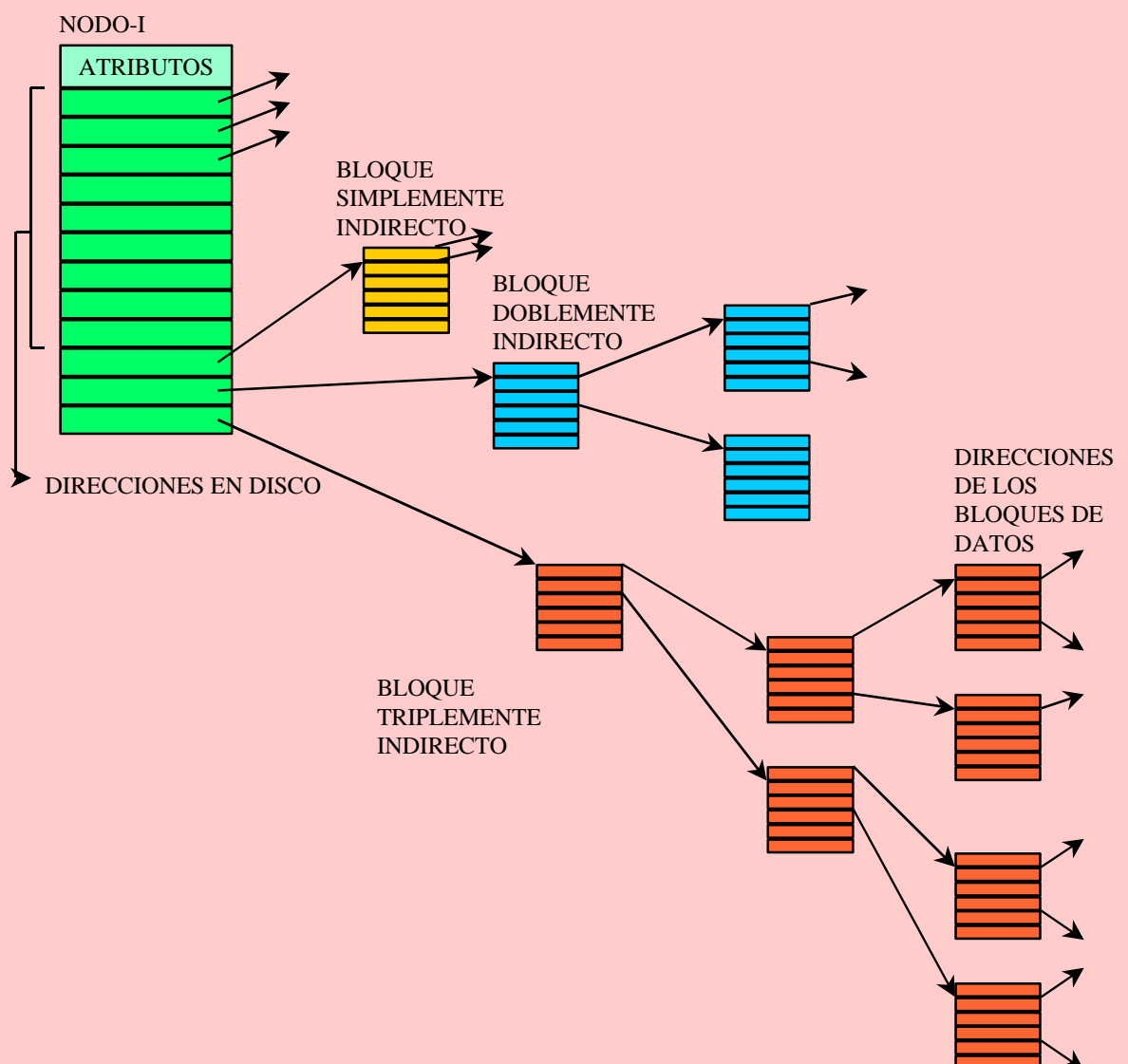
BLOQUE 0 B(4)	BLOQUE 1 B(10)	BLOQUE 2 C(1)	BLOQUE 3 A(4)	BLOQUE 4 B(8)	BLOQUE 5 C(2)	BLOQUE 6 B(1)
BLOQUE 7 LIBRE	BLOQUE 8 A(1)	BLOQUE 9 B(9)	BLOQUE 10 B(2)	BLOQUE 11 LIBRE	BLOQUE 12 A(3)	BLOQUE 13 B(7)
BLOQUE 14 B(3)	BLOQUE 15 LIBRE	BLOQUE 16 LIBRE	BLOQUE 17 A(2)	BLOQUE 18 B(6)	BLOQUE 19 C(5)	BLOQUE 20 C(3)
BLOQUE 21 LIBRE	BLOQUE 22 B(5)	BLOQUE 23 C(4)	BLOQUE 24 LIBRE	BLOQUE 25 LIBRE	BLOQUE 26 A(5)	BLOQUE 27 LIBRE

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- υ NODOS-I (NODOS INDICES):
- υ SE ASOCIA A CADA ARCHIVO UNA PEQUEÑA TABLA, LLAMADA **NODO-I (NODO INDICE)**:
 - Φ CONTIENE LOS ATRIBUTOS Y DIRECCIONES EN DISCO DE LOS BLOQUES DEL ARCHIVO.
 - Φ SE TRASLADA DEL DISCO A LA MEMORIA PRINCIPAL AL ABRIR EL ARCHIVO.
 - Φ EN RIGOR, ALMACENA SOLO LAS PRIMERAS DIRECCIONES EN DISCO:
 - SI EL ARCHIVO ES PEQUEÑO, TODA LA INFORMACION ESTA EN EL NODO-I.
 - SI EL ARCHIVO ES GRANDE, UNA DE LAS DIRECCIONES EN EL NODO-I ES LA DIRECCION DE UN BLOQUE EN EL DISCO LLAMADO **BLOQUE SIMPLEMENTE INDIRECTO**:
 - CONTIENE LAS DIRECCIONES EN DISCO ADICIONALES.
 - SI RESULTA INSUFICIENTE, OTRA DIRECCION EN EL NODO-I, EL **BLOQUE DOBLEMENTE INDIRECTO**, CONTIENE LA DIRECCION DE UN BLOQUE QUE PRESENTA UNA LISTA DE LOS BLOQUES SIMPLEMENTE INDIRECTOS:
 - CADA BLOQUE SIMPLEMENTE INDIRECTO APUNTA A UN GRUPO DE BLOQUES DE DATOS.
 - DE SER NECESARIO SE PUEDEN UTILIZAR **BLOQUES TRIPLEMENTE INDIRECTOS**.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

UN NODO-I



IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v **IMPLANTACION DE DIRECTORIOS:**
- v PARA ABRIR UN ARCHIVO EL S. O. UTILIZA INFORMACION DEL DIRECTORIO:
 - u EL DIRECTORIO CONTIENE LA INFORMACION NECESARIA PARA ENCONTRAR LOS BLOQUES EN EL DISCO.
 - u EL TIPO DE INFORMACION VARIA SEGUN EL SISTEMA.
- v LA PRINCIPAL FUNCION DEL SISTEMA DE DIRECTORIOS ES ASOCIAR EL NOMBRE DEL ARCHIVO CON LA INFORMACION NECESARIA PARA LOCALIZAR LOS DATOS.
- v UN ASPECTO INTIMAMENTE LIGADO CON ESTO ES LA POSICION DE ALMACENAMIENTO DE LOS ATRIBUTOS:
 - u UNA POSIBILIDAD ES ALMACENARLOS EN FORMA DIRECTA DENTRO DEL DATO DEL DIRECTORIO.
 - u OTRA POSIBILIDAD ES ALMACENAR LOS ATRIBUTOS EN EL NODO-I EN VEZ DE UTILIZAR LA ENTRADA DEL DIRECTORIO.
- v **ARCHIVOS COMPARTIDOS:**
- v FRECUENTEMENTE CONVIENE QUE LOS ARCHIVOS COMPARTIDOS APAREZCAN SIMULTANEAMENTE EN:
 - u DISTINTOS DIRECTORIOS DE DISTINTOS USUARIOS.
- v EL PROPIO SISTEMA DE ARCHIVOS ES UNA **GRAFICA DIRIGIDA ACICLICA** EN VEZ DE UN ARBOL.
- v LA CONEXION ENTRE UN DIRECTORIO Y UN ARCHIVO DE OTRO DIRECTORIO AL CUAL COMPARTEN SE DENOMINA **ENLACE**.
- v SI LOS DIRECTORIOS REALMENTE CONTIENEN DIRECCIONES EN DISCO:
 - u SE DEBE TENER UNA COPIA DE LAS DIRECCIONES EN DISCO EN EL DIRECTORIO QUE ACCEDE AL ARCHIVO COMPARTIDO AL **ENLAZAR** EL ARCHIVO.
 - u SE DEBE EVITAR QUE LOS CAMBIOS HECHOS POR UN USUARIO A TRAVES DE UN DIRECTORIO NO SEAN VISIBLES POR LOS DEMAS USUARIOS:
 - Φ SE CONSIDERARAN DOS SOLUCIONES POSIBLES.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

v PRIMER SOLUCION:

- u LOS BLOQUES DEL DISCO NO SE ENLISTAN EN LOS DIRECTORIOS, SINO EN UNA PEQUEÑA ESTRUCTURA DE DATOS ASOCIADA AL PROPIO ARCHIVO.
- u LOS DIRECTORIOS APUNTARIAN SOLO A ESA PEQUEÑA ESTRUCTURA DE DATOS, QUE PODRIA SER EL NODO-I.

v SEGUNDA SOLUCION:

- u EL ENLACE SE PRODUCE HACIENDO QUE EL SISTEMA CREE UN NUEVO ARCHIVO DE TIPO "LINK".
- u EL ARCHIVO "LINK":
 - ⊕ INGRESA AL DIRECTORIO DEL USUARIO QUE ACCEDE A UN ARCHIVO DE OTRO DIRECTORIO Y USUARIO.
 - ⊕ SOLO CONTIENE EL NOMBRE DE LA RUTA DE ACCESO DEL ARCHIVO AL CUAL SE ENLAZA.
- u ESTE CRITERIO SE DENOMINA **ENLACE SIMBOLICO**.

v DESVENTAJAS DE LA PRIMER SOLUCION:

- u LA CREACION DE UN ENLACE:
 - ⊕ NO MODIFICA LA PROPIEDAD RESPECTO DE UN ARCHIVO.
 - ⊕ AUMENTA EL CONTADOR DE ENLACES DEL NODO-I:
 - EL SISTEMA SABE EL N° DE ENTRADAS DE DIRECTORIO QUE APUNTAN EN CIERTO MOMENTO AL ARCHIVO.
- u SI EL PROPIETARIO INICIAL DEL ARCHIVO INTENTA ELIMINARLO, SURGE UN PROBLEMA PARA EL SISTEMA:
 - ⊕ SI ELIMINA EL ARCHIVO Y LIMPIA EL NODO-I, EL DIRECTORIO QUE ENLAZO AL ARCHIVO TENDRA UNA ENTRADA QUE APUNTA A UN NODO-I NO VALIDO.
 - ⊕ SI EL NODO-I SE REASIGNA A OTRO ARCHIVO EL ENLACE APUNTARA AL ARCHIVO INCORRECTO.
 - ⊕ EL SISTEMA:
 - PUEDE VER POR MEDIO DEL CONTADOR DE ENLACES EN EL NODO-I QUE EL ARCHIVO SIGUE UTILIZANDOSE.
 - NO PUEDE LOCALIZAR TODAS LAS ENTRADAS DE DIRECTORIO ASOCIADAS A ESE ARCHIVO PARA ELIMINARLAS.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

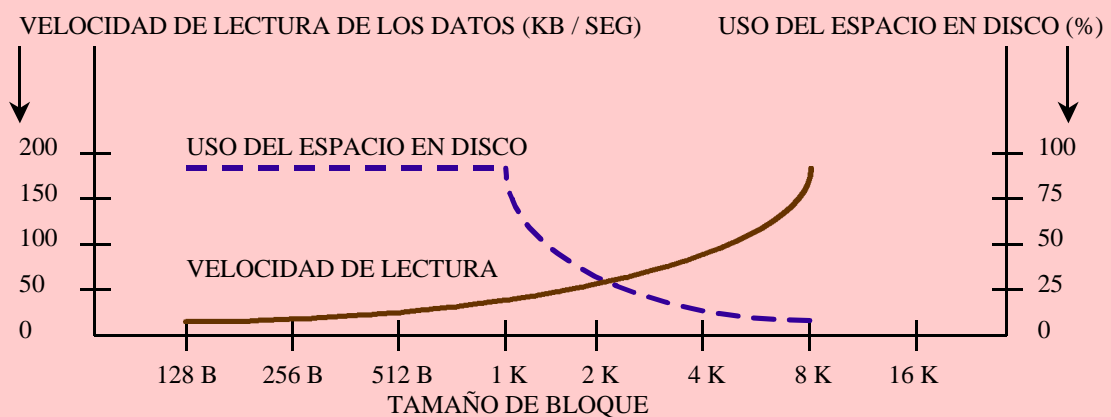
- u LA SOLUCION PODRIA SER:
 - Φ ELIMINAR LA ENTRADA DEL DIRECTORIO INICIALMENTE PROPIETARIO DEL ARCHIVO.
 - Φ DEJAR INTACTO EL NODO-I:
 - SE DARIA EL CASO QUE EL DIRECTORIO QUE POSEE EL ENLACE ES EL UNICO QUE POSEE UNA ENTRADA DE DIRECTORIO PARA UN ARCHIVO DE OTRO DIRECTORIO, PARA EL CUAL DICHO ARCHIVO YA NO EXISTE.
 - ESTO NO OCURRE CON LOS ENLACES SIMBOLICOS YA QUE SOLO EL PROPIETARIO VERDADERO TIENE UN APUNTADOR AL NODO-I:
 - LOS USUARIOS ENLAZADOS AL ARCHIVO SOLO TIENEN NOMBRES DE RUTAS DE ACCESO Y NO APUNTADES A NODO-I.
 - CUANDO EL PROPIETARIO ELIMINA UN ARCHIVO, ESTE SE DESTRUYE.
- v DESVENTAJAS DE LA SEGUNDA SOLUCION:
 - u EL PRINCIPAL PROBLEMA ES SU COSTO EXCESIVO, ESPECIALMENTE EN ACCESOS A DISCO:
 - Φ SE DEBE LEER EL ARCHIVO QUE CONTIENE LA RUTA DE ACCESO, ANALIZARLA Y SEGUIRLA COMPONENTE A COMPONENTE HASTA ALCANZAR EL NODO-I.
 - u SE PRECISA UN NODO-I ADICIONAL POR CADA ENLACE SIMBOLICO Y UN BLOQUE ADICIONAL EN DISCO PARA ALMACENAR LA RUTA DE ACCESO.
 - u LOS ARCHIVOS PUEDEN TENER DOS O MAS RUTAS DE ACCESO:
 - Φ EN BUSQUEDAS GENERICAS SE PODRIA ENCONTRAR EL MISMO ARCHIVO POR DISTINTAS RUTAS Y TRATARSELO COMO SI FUERAN ARCHIVOS DISTINTOS.
- v LOS ENLACES SIMBOLICOS TIENEN LA VENTAJA DE QUE SE PUEDEN UTILIZAR PARA ENLAZAR ARCHIVOS EN OTRAS MAQUINAS, EN CUALQUIER PARTE DEL MUNDO:
 - u SE DEBE PROPORCIONAR SOLO LA DIRECCION DE LA RED DE LA MAQUINA DONDE RESIDE EL ARCHIVO Y SU RUTA DE ACCESO EN ESA MAQUINA.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v **ADMINISTRACION DEL ESPACIO EN DISCO:**
- v EXISTEN DOS ESTRATEGIAS GENERALES PARA ALMACENAR UN ARCHIVO DE “N” BYTES:
 - u ASIGNAR “N” BYTES CONSECUTIVOS DE ESPACIO EN EL DISCO:
 - Φ TIENE EL PROBLEMA DE QUE SI UN ARCHIVO CRECE SERA MUY PROBABLE QUE DEBA DESPLAZARSE EN EL DISCO:
 - PUEDE AFECTAR SERIAMENTE AL RENDIMIENTO.
 - u DIVIDIR EL ARCHIVO EN CIERTO N° DE BLOQUES (NO NECESARIAMENTE) ADYACENTES:
 - Φ GENERALMENTE LOS SISTEMAS DE ARCHIVOS UTILIZAN ESTA ESTRATEGIA CON BLOQUES DE TAMAÑO FIJO.
- v TAMAÑO DEL BLOQUE:
- v DADA LA FORMA EN QUE ESTAN ORGANIZADOS LOS BLOQUES, EL SECTOR, LA PISTA Y EL CILINDRO CON LOS CANDIDATOS OBVIOS COMO UNIDADES DE ASIGNACION.
- v SI SE TIENE UNA UNIDAD DE ASIGNACION GRANDE, COMO UN CILINDRO, ESTO SIGNIFICA QUE CADA ARCHIVO, INCLUSIVE UNO PEQUEÑO, OCUPARA TODO UN CILINDRO:
 - u SE DESPERDICIA ESPACIO DE ALMACENAMIENTO EN DISCO.
- v SI SE UTILIZA UNA UNIDAD DE ASIGNACION PEQUEÑA, COMO UN SECTOR, IMPLICA QUE CADA ARCHIVO CONSTARA DE MUCHOS BLOQUES:
 - u SU LECTURA GENERARA MUCHAS OPERACIONES DE E / S AFECTANDO LA PERFORMANCE.
- v LA EFICIENCIA EN TIEMPO Y ESPACIO TIENEN UN CONFLICTO INHERENTE.
- v GENERALMENTE SE UTILIZAN COMO SOLUCION DE COMPROMISO BLOQUES DE 1/2 K, 1K, 2K O 4K.
- v HAY QUE RECORDAR QUE EL TIEMPO DE LECTURA DE UN BLOQUE DE DISCO ES LA SUMA DE LOS TIEMPOS DE:
 - u BUSQUEDA.
 - u DEMORA ROTACIONAL.
 - u TRANSFERENCIA.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

REPRESENTACION DE LA VELOCIDAD DE LECTURA Y DEL USO DEL ESPACIO EN DISCO EN FUNCION DEL TAMAÑO DE BLOQUE



- v REGISTRO DE LOS BLOQUES LIBRES:
- v SE UTILIZAN POR LO GENERAL DOS METODOS:
 - v LA LISTA DE BLOQUES LIBRES COMO LISTA LIGADA.
 - v UN MAPA DE BITS.
- v LISTA LIGADA DE BLOQUES DE DISCO:
 - v CADA BLOQUE CONTIENE TANTOS N° DE BLOQUES LIBRES COMO PUEDA.
 - v LOS BLOQUES LIBRES SE UTILIZAN PARA CONTENER A LA LISTA DE BLOQUES LIBRES.
- v MAPA DE BITS:
 - v UN DISCO CON “N” BLOQUES NECESITA UN MAPA DE BITS CON “N” BITS.
 - v LOS BLOQUES LIBRES SE REPRESENTA CON “1” Y LOS ASIGNADOS CON “0” (O VICEVERSA).
 - v GENERALMENTE ESTE METODO ES PREFERIBLE CUANDO EXISTE ESPACIO SUFICIENTE EN LA MEMORIA PRINCIPAL PARA CONTENER COMPLETO EL MAPA DE BITS.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v DISK QUOTAS:
- v PARA EVITAR QUE LOS USUARIOS SE APROPIEN DE UN ESPACIO EXCESIVO EN DISCO, LOS S. O. MULTIUSUARIO PROPORCIONAN GENERALMENTE UN MECANISMO PARA ESTABLECER LAS CUOTAS EN EL DISCO.
- v LA IDEA ES QUE:
 - u UN ADMINISTRADOR DEL SISTEMA ASIGNE A CADA USUARIO UNA PROPORCION MAXIMA DE ARCHIVOS Y BLOQUES.
 - u EL S. O. GARANTICE QUE LOS USUARIOS NO EXCEDAN SUS CUOTAS.
- v UN MECANISMO UTILIZADO ES EL SIGUIENTE:
 - u CUANDO UN USUARIO ABRE UN ARCHIVO:
 - ⦿ SE LOCALIZAN LOS ATRIBUTOS Y DIRECCIONES EN DISCO.
 - ⦿ SE COLOCAN EN UNA TABLA DE ARCHIVOS ABIERTOS EN LA MEMORIA PRINCIPAL.
 - ⦿ UNO DE LOS ATRIBUTOS INDICA EL PROPIETARIO DEL ARCHIVO:
 - CUALQUIER AUMENTO DEL TAMAÑO DEL ARCHIVO SE CARGA A LA CUOTA DEL PROPIETARIO.
 - ⦿ UNA SEGUNDA TABLA CONTIENE EL REGISTRO DE LAS CUOTAS PARA C / U DE LOS USUARIOS QUE TENGAN UN ARCHIVO ABIERTO EN ESE MOMENTO:
 - AUN CUANDO EL ARCHIVO LO HAYA ABIERTO OTRO USUARIO.
 - u CUANDO SE ESCRIBE UNA NUEVA ENTRADA EN LA TABLA DE ARCHIVOS ABIERTOS:
 - ⦿ SE INTRODUCE UN APUNTADOR AL REGISTRO DE LA CUOTA DEL PROPIETARIO PARA LOCALIZAR LOS LIMITES.
 - u CUANDO SE AÑADE UN BLOQUE A UN ARCHIVO:
 - ⦿ SE INCREMENTA EL TOTAL DE BLOQUES CARGADOS AL PROPIETARIO.
 - ⦿ SE VERIFICA ESTE VALOR CONTRA LOS LIMITES ESTRICTO Y FLEXIBLE (EL 1º NO SE PUEDE SUPERAR, EL 2º SI).
 - ⦿ TAMBIEN SE VERIFICA EL N° DE ARCHIVOS.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v **CONFIABILIDAD DEL SISTEMA DE ARCHIVOS:**
- v ES NECESARIO PROTEGER LA INFORMACION ALOJADA EN EL SISTEMA DE ARCHIVOS, EFECTUANDO LOS RESGUARDOS CORRESPONDIENTES.
- v DE ESTA MANERA SE EVITAN LAS CONSECUENCIAS GENERALMENTE CATASTROFICAS DE LA PERDIDA DE LOS SISTEMAS DE ARCHIVOS.
- v LAS PERDIDAS SE PUEDEN DEBER A PROBLEMAS DE HARDWARE, SOFTWARE, HECHOS EXTERNOS, ETC.
- v MANEJO DE UN BLOQUE DEFECTUOSO:
- v SE UTILIZAN SOLUCIONES POR HARDWARE Y POR SOFTWARE.
- v LA SOLUCION EN HARDWARE:
 - u CONSISTE EN DEDICAR UN SECTOR DEL DISCO A LA LISTA DE BLOQUES DEFECTUOSOS.
 - u AL INICIALIZAR EL CONTROLADOR POR PRIMERA VEZ:
 - Φ LEE LA “LISTA DE BLOQUES DEFECTUOSOS”.
 - Φ ELIGE UN BLOQUE (O PISTA) DE RESERVA PARA REEMPLAZAR LOS DEFECTUOSOS.
 - Φ REGISTRA LA ASOCIACION EN LA LISTA DE BLOQUES DEFECTUOSOS.
 - Φ EN LO SUCESIVO, LAS SOLICITUDES DEL BLOQUE DEFECTUOSO UTILIZARAN EL DE REPUESTO.
- v LA SOLUCION EN SOFTWARE:
 - u REQUIERE QUE EL USUARIO O EL SISTEMA DE ARCHIVOS CONSTRUYAN UN ARCHIVO CON TODOS LOS BLOQUES DEFECTUOSOS.
 - u SE LOS ELIMINA DE LA “LISTA DE BLOQUES LIBRES”.
 - u SE CREA UN “ARCHIVO DE BLOQUES DEFECTUOSOS”:
 - Φ ESTA CONSTITUIDO POR LOS BLOQUES DEFECTUOSOS.
 - Φ NO DEBE SER LEIDO NI ESCRITO.
 - Φ NO SE DEBE INTENTAR OBTENER COPIAS DE RESPALDO DE ESTE ARCHIVO.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v RESPALDOS (COPIAS DE SEGURIDAD O DE BACK-UP):
- v ES MUY IMPORTANTE RESPALDAR LOS ARCHIVOS CON FRECUENCIA.
- v LOS RESPALDOS PUEDEN CONSISTIR EN EFECTUAR **COPIAS COMPLETAS** DEL CONTENIDO DE LOS DISCOS (FLEXIBLES O RIGIDOS).
- v UNA ESTRATEGIA DE RESPALDO CONSISTE EN **DIVIDIR LOS DISCOS EN AREAS DE DATOS Y AREAS DE RESPALDO**, UTILIZANDOLAS DE A PARES:
 - u SE DESPERDICIA LA MITAD DEL ALMACENAMIENTO DE DATOS EN DISCO PARA RESPALDO.
 - u CADA NOCHE (O EN EL MOMENTO QUE SE ESTABLEZCA), LA PARTE DE DATOS DE LA UNIDAD 0 SE COPIA A LA PARTE DE RESPALDO DE LA UNIDAD 1 Y VICEVERSA.
- v OTRA ESTRATEGIA ES EL **VACIADO POR INCREMENTOS O RESPALDO INCREMENTAL**:
 - u SE OBTIENE UNA COPIA DE RESPALDO PERIODICAMENTE (POR EJ. UNA VEZ POR MES O POR SEMANA), LLAMADA COPIA TOTAL.
 - u SE OBTIENE UNA COPIA DIARIA SOLO DE AQUELLOS ARCHIVOS MODIFICADOS DESDE LA ULTIMA COPIA TOTAL:
 - Φ EN ESTRATEGIAS MEJORADAS, SE COPIAN SOLO AQUELLOS ARCHIVOS MODIFICADOS DESDE LA ULTIMA VEZ QUE DICHOS ARCHIVOS FUERON COPIADOS.
 - u SE DEBE MANTENER EN EL DISCO INFORMACION DE CONTROL COMO UNA “LISTA DE LOS TIEMPOS DE COPIADO” DE CADA ARCHIVO:
 - Φ DEBE SER ACTUALIZADA CADA VEZ QUE SE OBTIENEN COPIAS DE LOS ARCHIVOS Y CADA VEZ QUE LOS ARCHIVOS SON MODIFICADOS.
 - u PUEDE REQUERIR UNA GRAN CANTIDAD DE CINTAS DE RESPALDO DEDICADAS A LOS RESPALDOS DIARIOS ENTRE RESPALDOS COMPLETOS.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v CONSISTENCIA DEL SISTEMA DE ARCHIVOS:
- v MUCHOS SISTEMAS DE ARCHIVOS LEEN BLOQUES, LOS MODIFICAN Y ESCRIBEN EN ELLOS DESPUES.
- v SI EL SISTEMA FALLA ANTES DE ESCRIBIR EN LOS BLOQUES MODIFICADOS, EL SISTEMA DE ARCHIVOS PUEDE QUEDAR EN UN “ESTADO INCONSISTENTE”.
- v LA INCONSISTENCIA ES PARTICULARMENTE CRITICA SI ALGUNO DE LOS BLOQUES AFECTADOS SON:
 - u BLOQUES DE NODOS-I.
 - u BLOQUES DE DIRECTORIOS.
 - u BLOQUES DE LA LISTA DE BLOQUES LIBRES.
- v LA MAYORIA DE LOS SISTEMAS DISPONE DE UN PROGRAMA UTILITARIO QUE VERIFICA LA CONSISTENCIA DEL SISTEMA DE ARCHIVOS:
 - u SE PUEDEN EJECUTAR AL ARRANCAR EL SISTEMA O A PEDIDO.
 - u PUEDEN ACTUAR SOBRE TODOS O ALGUNOS DE LOS DISCOS.
 - u PUEDEN EFECTUAR VERIFICACIONES A NIVEL DE BLOQUES Y A NIVEL DE ARCHIVOS.
 - u LA CONSISTENCIA DEL SISTEMA DE ARCHIVOS NO ASEGURA LA CONSISTENCIA INTERNA DE CADA ARCHIVO, RESPECTO DE SU CONTENIDO.
 - u GENERALMENTE PUEDEN VERIFICAR TAMBIEN EL SISTEMA DE DIRECTORIOS Y/O DE BIBLIOTECAS.
- v GENERALMENTE LOS UTILITARIOS UTILIZAN DOS TABLAS:
 - u TABLA DE BLOQUES EN USO.
 - u TABLA DE BLOQUES LIBRES.
 - u CADA BLOQUE DEBE ESTAR REFERENCIADO EN UNA DE ELLAS.
- v SI UN BLOQUE NO APARECE EN NINGUNA DE LAS TABLAS SE TRATA DE UNA FALLA LLAMADA **BLOQUE FALTANTE**:
 - u NO PRODUCE DAÑOS PERO DESPERDICIA ESPACIO EN DISCO.
 - u SE SOLUCIONA AÑADIENDO EL BLOQUE A LA TABLA DE BLOQUES LIBRES.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v TAMBIEN PODRIA DETECTARSE LA SITUACION DE FALLA DEBIDA A UN BLOQUE REFERENCIADO DOS VECES EN LA TABLA DE BLOQUES LIBRES:
 - u ESTA FALLA NO SE PRODUCE EN LOS SISTEMAS DE ARCHIVOS BASADOS EN MAPAS DE BITS, SI EN LOS BASADOS EN TABLAS O LISTAS.
 - u LA SOLUCION CONSISTE EN DEPURAR LA TABLA DE BLOQUES LIBRES.
- v UNA FALLA MUY GRAVE ES QUE EL MISMO BLOQUE DE DATOS APAREZCA REFERENCIADO DOS O MAS VECES EN LA TABLA DE BLOQUES EN USO:
 - u COMO PARTE DEL MISMO O DE DISTINTOS ARCHIVOS.
 - u SI UNO DE LOS ARCHIVOS SE BORRA, EL BLOQUE APARECERIA EN LA TABLA DE BLOQUES LIBRES Y TAMBIEN EN LA DE BLOQUES EN USO.
 - u UNA SOLUCION ES QUE EL VERIFICADOR DEL SISTEMA DE ARCHIVOS:
 - ❖ ASIGNE UN BLOQUE LIBRE.
 - ❖ COPIE EN EL BLOQUE LIBRE EL CONTENIDO DEL BLOQUE CONFLICTIVO.
 - ❖ ACTUALICE LAS TABLAS AFECTANDO EL BLOQUE COPIA A ALGUNO DE LOS ARCHIVOS.
 - ❖ AGREGUE EL BLOQUE CONFLICTIVO A LA TABLA DE BLOQUES LIBRES.
 - ❖ INFORME AL USUARIO PARA QUE VERIFIQUE EL DAÑO DETECTADO Y LA SOLUCION DADA.
- v OTRO ERROR POSIBLE ES QUE UN BLOQUE ESTE EN LA TABLA DE BLOQUES EN USO Y EN LA TABLA DE BLOQUES LIBRES:
 - u SE SOLUCIONA ELIMINANDOLO DE LA TABLA DE BLOQUES LIBRES.
- v LAS VERIFICACIONES DE DIRECTORIOS INCLUYEN CONTROLES COMO:
 - u N° DE DIRECTORIOS QUE APUNTAN A UN NODO-I CON LOS CONTADORES DE ENLACES ALMACENADOS EN LOS PROPIOS NODOS-I:
 - ❖ EN UN SISTEMA CONSISTENTE DE ARCHIVOS DEBEN

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v UNA POSIBLE FALLA ES QUE EL CONTADOR DE ENLACES SEA MAYOR QUE EL N° DE ENTRADAS DEL DIRECTORIO:
 - u AUNQUE SE ELIMINARAN TODOS LOS ARCHIVOS DE LOS DIRECTORIOS EL CONTADOR SERIA DISTINTO DE CERO Y NO SE PODRIA ELIMINAR EL NODO-I.
 - u NO SE TRATA DE UN ERROR SERIO PERO PRODUCE DESPERDICIO DE ESPACIO EN DISCO CON ARCHIVOS QUE NO SE ENCUENTRAN EN NINGUN DIRECTORIO.
 - u SE SOLUCIONA HACIENDO QUE EL CONTADOR DE ENLACES EN EL NODO-I TOMA EL VALOR CORRECTO:
 - Φ SI EL VALOR CORRECTO ES 0, EL ARCHIVO DEBE ELIMINARSE.
- v OTRO TIPO DE ERROR ES POTENCIALMENTE CATASTROFICO:
 - u SI DOS ENTRADAS DE UN DIRECTORIO SE ENLAZAN A UN ARCHIVO, PERO EL NODO-I INDICA QUE SOLO EXISTE UN ENLACE, ENTONCES, AL ELIMINAR CUALQUIERA DE ESTAS ENTRADAS DE DIRECTORIO, EL CONTADOR DEL NODO-I TOMARA EL VALOR 0.
 - u DEBIDO AL VALOR 0 EL SISTEMA DE ARCHIVOS LO SEÑALA COMO NO UTILIZADO Y LIBERA TODOS SUS BLOQUES.
 - u UNO DE LOS DIRECTORIOS APUNTA HACIA UN NODO-I NO UTILIZADO, CUYOS BLOQUES SE PODRIAN ASIGNAR ENTONCES A OTROS ARCHIVOS.
 - u LA SOLUCION ES FORZAR QUE EL CONTADOR DE ENLACES DEL NODO-I SEA IGUAL AL N° DE ENTRADAS DEL DIRECTORIO.
- v TAMBIEN SE PUEDEN HACER VERIFICACIONES HEURISTICAS, POR EJ.:
 - u CADA NODO-I TIENE UN MODO, PERO ALGUNOS MODOS SON VALIDOS AUNQUE EXTRAÑOS:
 - Φ EJ.: SE PROHIBE EL ACCESO AL PROPIETARIO Y TODO SU GRUPO, PERO SE PERMITE A LOS EXTRAÑOS LEER, ESCRIBIR Y EJECUTAR EL ARCHIVO.
 - Φ LA VERIFICACION DEBERIA DETECTAR E INFORMAR DE ESTAS SITUACIONES.
 - u SE DEBERIA INFORMAR COMO SOSPECHOSOS AQUELLOS

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v **DESEMPEÑO DEL SISTEMA DE ARCHIVOS:**
- v EL ACCESO AL DISCO ES MUCHO MAS LENTO QUE EL ACCESO A LA MEMORIA:
 - u LOS TIEMPOS SE MIDEN EN MILISEGUNDOS Y EN NANOSEGUNDOS RESPECTIVAMENTE.
 - u SE DEBE REDUCIR EL N° DE ACCESOS A DISCO.
- v LA TECNICA MAS COMUN PARA REDUCIR LOS ACCESOS A DISCO ES EL **BLOQUE CACHE O BUFFER CACHE:**
 - u SE UTILIZA EL TERMINO **OCULTAMIENTO** PARA ESTA TECNICA:
 - Φ DEL FRANCES “CACHER”: OCULTAR.
 - u UN CACHE ES UNA COLECCION DE BLOQUES QUE PERTENECEN DESDE EL PUNTO DE VISTA LOGICO AL DISCO, PERO QUE SE MANTIENEN EN MEMORIA POR RAZONES DE RENDIMIENTO.
- v UNO DE LOS ALGORITMOS MAS COMUNES PARA LA ADMINISTRACION DEL CACHE ES EL SIGUIENTE:
 - u VERIFICAR TODAS LAS SOLICITUDES DE LECTURA PARA SABER SI EL BLOQUE SOLICITADO SE ENCUENTRA EN EL CACHE.
 - u EN CASO AFIRMATIVO, SE SATISFACE LA SOLICITUD SIN UN ACCESO A DISCO.
 - u EN CASO NEGATIVO, SE LEE PARA QUE INGRESE AL CACHE Y LUEGO SE COPIA AL LUGAR DONDE SE NECESITE.
 - u CUANDO HAY QUE CARGAR UN BLOQUE EN UN CACHE TOTALMENTE OCUPADO:
 - Φ HAY QUE ELIMINAR ALGUN BLOQUE Y VOLVERLO A ESCRIBIR EN EL DISCO EN CASO DE QUE HAYA SIDO MODIFICADO LUEGO DE HABERLO TRAIIDO DEL DISCO.
 - Φ SE PLANTEA UNA SITUACION MUY PARECIDA A LA PAGINACION Y SE RESUELVE CON ALGORITMOS SIMILARES.
- v SE DEBE CONSIDERAR LA POSIBILIDAD DE UNA FALLA TOTAL DEL SISTEMA Y SU IMPACTO EN LA CONSISTENCIA DEL SISTEMA DE ARCHIVOS:
 - u SI UN BLOQUE CRITICO, COMO UN BLOQUE DE UN NODO-I, SE LEE EN EL CACHE Y SE MODIFICA, SIN VOLVERSE A ESCRIBIR EN EL DISCO, UNA FALLA TOTAL DEL SISTEMA DEJARÁ AL SISTEMA DE ARCHIVOS EN UN ESTADO INCONSISTENTE.

IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO

- v SE DEBEN TENER EN CUENTA LOS SIGUIENTES FACTORES:
 - u ¿ ES POSIBLE QUE EL BLOQUE MODIFICADO SE VUELVA A NECESITAR MUY PRONTO ?:
 - Φ LOS BLOQUES QUE SE VAYAN A UTILIZAR MUY PRONTO, COMO UN BLOQUE PARCIALMENTE OCUPADO QUE SE ESTA ESCRIBIENDO, DEBERIAN PERMANECER UN “LARGO TIEMPO”.
 - u ¿ ES ESENCIAL EL BLOQUE PARA LA CONSISTENCIA DEL SISTEMA DE ARCHIVOS ?:
 - Φ SI ES ESENCIAL (GENERALMENTE LO SERA SI NO ES BLOQUE DE DATOS) Y HA SIDO MODIFICADO, DEBE ESCRIBIRSE EN EL DISCO DE INMEDIATO:
 - SE REDUCE LA PROBABILIDAD DE QUE UNA FALLA TOTAL DEL SISTEMA HAGA NAUFRAGAR AL SISTEMA DE ARCHIVOS.
 - SE DEBE ELEGIR CON CUIDADO EL ORDEN DE ESCRITURA DE LOS BLOQUES CRITICOS.
 - u NO ES RECOMENDABLE MANTENER LOS BLOQUES DE DATOS EN EL CACHE DURANTE MUCHO TIEMPO ANTES DE REESCRIBIRLOS.
- v LA SOLUCION DE ALGUNOS S. O. CONSISTE EN TENER UNA LLAMADA AL SISTEMA QUE FUERZA UNA ACTUALIZACION GENERAL A INTERVALOS REGULARES DE ALGUNOS SEGUNDOS (POR EJ. 30).
- v OTRA SOLUCION CONSISTE EN ESCRIBIR LOS BLOQUES MODIFICADOS AL DISCO, TAN PRONTO COMO HAYA SIDO ESCRITO:
 - u SE DICE QUE SE TRATA DE **CACHES DE ESCRITURA**.
 - u REQUIERE MAS E / S QUE OTROS TIPOS DE CACHES.
- v UNA TECNICA IMPORTANTE PARA AUMENTAR EL RENDIMIENTO DE UN SISTEMA DE ARCHIVOS ES LA REDUCCION DE LA CANTIDAD DE MOVIMIENTOS DEL BRAZO DEL DISCO (MECANISMO DE ACCESO):
 - u SE DEBEN COLOCAR LOS BLOQUES QUE PROBABLEMENTE TENGAN UN ACCESO SECUENCIAL, PROXIMOS ENTRE SI, PREFERENTEMENTE EN EL MISMO CILINDRO.
 - u LOS NODOS-I DEBEN ESTAR A MITAD DEL DISCO Y NO AL PRINCIPIO, REDUCIENDO A LA MITAD EL TIEMPO PROMEDIO DE

DESCRIPTOR DE ARCHIVOS

- v EL **DESCRIPTOR DE ARCHIVOS O BLOQUE DE CONTROL DE ARCHIVOS** ES UN BLOQUE DE CONTROL QUE CONTIENE INFORMACION QUE EL SISTEMA NECESITA PARA ADMINISTRAR UN ARCHIVO.
- v ES UNA ESTRUCTURA MUY DEPENDIENTE DEL SISTEMA.
- v PUEDE INCLUIR LA SIGUIENTE INFORMACION:
 - u NOMBRE SIMBOLICO DEL ARCHIVO.
 - u LOCALIZACION DEL ARCHIVO EN EL ALMACENAMIENTO SECUNDARIO.
 - u ORGANIZACION DEL ARCHIVO (METODO DE ORGANIZACION Y ACCESO).
 - u TIPO DE DISPOSITIVO.
 - u DATOS DE CONTROL DE ACCESO.
 - u TIPO (ARCHIVO DE DATOS, PROGRAMA OBJETO, PROGRAMA FUENTE, ETC.).
 - u DISPOSICION (PERMANENTE CONTRA TEMPORAL).
 - u FECHA Y TIEMPO DE CREACION.
 - u FECHA DE DESTRUCCION.
 - u FECHA DE LA ULTIMA MODIFICACION.
 - u SUMA DE LAS ACTIVIDADES DE ACCESO (Nº DE LECTURAS, POR EJEMPLO).
- v LOS DESCRIPTORES DE ARCHIVOS SUELEN MANTENERSE EN EL ALMACENAMIENTO SECUNDARIO:
 - u SE PASAN AL ALMACENAMIENTO PRIMARIO AL ABRIR EL ARCHIVO.
- v EL DESCRIPTOR DE ARCHIVOS ES CONTROLADO POR EL SISTEMA DE ARCHIVOS:
 - u EL USUARIO PUEDE NO HACER REFERENCIA DIRECTA A EL.

SEGURIDAD

- v LOS SISTEMAS DE ARCHIVOS GENERALMENTE CONTIENEN INFORMACION MUY VALIOSA PARA SUS USUARIOS:
 - u LOS SISTEMAS DE ARCHIVOS DEBEN PROTEGERLA.
- v **EL AMBIENTE DE SEGURIDAD.**
- v SE ENTENDERA POR **SEGURIDAD** A LOS PROBLEMAS GENERALES RELATIVOS A LA GARANTIA DE QUE LOS ARCHIVOS NO SEAN LEIDOS O MODIFICADOS POR PERSONAL NO AUTORIZADO:
 - u INCLUYE ASPECTOS TECNICOS, DE ADMINISTRACION, LEGALES Y POLITICOS.
- v SE CONSIDERARAN **MECANISMOS DE PROTECCION** A LOS MECANISMOS ESPECIFICOS DEL SISTEMA OPERATIVO UTILIZADOS PARA RESGUARDAR LA INFORMACION DE LA COMPUTADORA.
- v LA FRONTERA ENTRE SEGURIDAD Y MECANISMOS DE PROTECCION NO ESTA BIEN DEFINIDA.
- v DOS DE LAS MAS IMPORTANTES FACETAS DE LA SEGURIDAD SON:
 - u LA PERDIDA DE DATOS.
 - u LOS INTRUSOS.
- v ALGUNAS DE LAS CAUSAS MAS COMUNES DE LA **PERDIDA DE DATOS** SON:
 - u ACTOS Y HECHOS DIVERSOS:
 - Φ INCENDIOS, INUNDACIONES, TERREMOTOS, GUERRAS, REVOLUCIONES, ROEDORES, ETC.
 - u ERRORES DE HARDWARE O DE SOFTWARE:
 - Φ FALLAS EN LA CPU, DISCOS O CINTAS ILEGIBLES, ERRORES DE TELECOMUNICACION, ERRORES EN LOS PROGRAMAS, ETC.
 - u ERRORES HUMANOS:
 - Φ ENTRADA INCORRECTA DE DATOS, MAL MONTAJE DE CINTAS O DISCOS, EJECUCION INCORRECTA DE PROGRAMAS, PERDIDA DE CINTAS O DISCOS, ETC.
- v LA MAYORIA DE ESTAS CAUSAS SE PUEDEN ENFRENTAR CON EL MANTENIMIENTO DE LOS RESPALDOS (BACK-UPS) ADECUADOS:
 - u DEBERIA HABER COPIAS EN UN LUGAR ALEJADO DE LOS DATOS

SEGURIDAD

- v RESPECTO DEL PROBLEMA DE LOS **INTRUSOS**, SE LOS PUEDE CLASIFICAR COMO:
 - u **PASIVOS**: SOLO DESEAN LEER ARCHIVOS QUE NO ESTAN AUTORIZADOS A LEER.
 - u **ACTIVOS**: DESEAN HACER CAMBIOS NO AUTORIZADOS A LOS DATOS.
- v PARA DISEÑAR UN SISTEMA SEGURO CONTRA INTRUSOS:
 - u HAY QUE TENER EN CUENTA EL TIPO DE INTRUSOS CONTRA LOS QUE SE DESEA TENER PROTECCION.
 - u HAY QUE SER CONSCIENTE DE QUE LA CANTIDAD DE ESFUERZO QUE SE PONE EN LA SEGURIDAD Y LA PROTECCION DEPENDE CLARAMENTE DE QUIEN SE PIENSA SEA EL ENEMIGO.
- v ALGUNOS TIPOS DE INTRUSOS SON LOS SIGUIENTES:
 - u CURIOSIDAD CASUAL DE USUARIOS NO TECNICOS.
 - u CONOCIDOS (TECNICAMENTE CAPACITADOS) HUSMEANDO.
 - u INTENTOS DELIBERADOS POR HACER DINERO.
 - u ESPIONAJE COMERCIAL O MILITAR.
- v OTRO ASPECTO DEL PROBLEMA DE LA SEGURIDAD ES LA **PRIVACIA**:
 - u PROTECCION DE LAS PERSONAS RESPECTO DEL MAL USO DE LA INFORMACION EN CONTRA DE UNO MISMO.
 - u IMPLICA ASPECTOS LEGALES Y MORALES.
- v TAMBIEN DEBE SEÑALARSE LA POSIBILIDAD DEL **ATAQUE DEL CABALLO DE TROYA**:
 - u MODIFICAR UN PROGRAMA NORMAL PARA QUE HAGA COSAS ADVERSAS ADEMAS DE SU FUNCION USUAL.
 - u ARREGLAR LAS COSAS PARA QUE LA VICTIMA UTILICE LA VERSION MODIFICADA.
- v ADEMAS DEBE CONSIDERARSE LA POSIBILIDAD DE ATAQUES AL ESTILO DEL **GUSANO DE INTERNET**:
 - u FUE LIBERADO POR ROBERT TAPPAN MORRIS EL 02/11/88 E HIZO QUE SE BLOQUEARAN LA MAYORIA DE LOS SISTEMAS SUN Y VAX DE INTERNET (FUE DESCUBIERTO Y CONDENADO).
 - u CONSTABA DE UN PROGRAMA ARRANCADOR Y DEL GUSANO PROPIAMENTE DICHO.
 - u UTILIZABA FALLAS DE SEGURIDAD DEL UNIX Y DE LOS

SEGURIDAD

- v UNA FORMA DE PROBAR LA SEGURIDAD DE UN SISTEMA ES CONTRATAR UN GRUPO DE EXPERTOS EN SEGURIDAD, CONOCIDO COMO EL **EQUIPO TIGRE O EQUIPO DE PENETRACION**:
 - u SU OBJETIVO ES INTENTAR PENETRAR EL SISTEMA DE SEGURIDAD PARA DESCUBRIR SUS FALENCIAS Y PROPONER SOLUCIONES.
- v OTRO ASPECTO IMPORTANTE DE LA SEGURIDAD CONSISTE EN NO SUBESTIMAR LOS PROBLEMAS QUE PUEDE CAUSAR EL PERSONAL.
- v **VIRUS.**
- v LOS VIRUS COMPUTACIONALES:
 - u CONSTITUYEN UNA CATEGORIA ESPECIAL DE ATAQUE.
 - u SON UN ENORME PROBLEMA PARA MUCHOS USUARIOS.
 - u SON FRAGMENTOS DE PROGRAMAS QUE SE AÑADEN A PROGRAMAS LEGITIMOS CON LA INTENCION DE INFECTAR A OTROS.
 - u UN VIRUS DIFIERE DE UN GUSANO EN LO SIGUIENTE:
 - ⦿ UN VIRUS ESTA A CUESTAS DE UN PROGRAMA EXISTENTE.
 - ⦿ UN GUSANO ES UN PROGRAMA COMPLETO EN SI MISMO.
 - u LOS VIRUS Y LOS GUSANOS INTENTAN DISEMINARSE Y PUEDEN CREAR UN DAÑO SEVERO.
 - u GENERALMENTE SE PROPAGAN A TRAVES DE COPIAS ILEGITIMAS DE PROGRAMAS.
 - u COMUNMENTE LOS VIRUS SE EJECUTAN E INTENTAN REPRODUCIRSE CADA VEZ QUE SE EJECUTA EL PROGRAMA QUE LOS ALOJA.
 - u FRECUENTEMENTE LOS PROBLEMAS CON LOS VIRUS SON MAS FACILES DE EVITAR QUE DE CURAR:
 - ⦿ UTILIZAR SOFTWARE ORIGINAL ADQUIRIDO EN COMERCIOS RESPETABLES.
 - ⦿ NO UTILIZAR COPIAS “PIRATAS”.
 - ⦿ EFECTUAR CONTROLES RIGUROSOS Y FRECUENTES CON PROGRAMAS ANTIVIRUS ACTUALIZADOS.
 - ⦿ TRABAJAR CON METODOLOGIA Y DISCIPLINA RIGUROSA EN EL INTERCAMBIO DE DISCOS Y EN LAS COPIAS A TRAVES DE REDES DE COMUNICACION DE DATOS.

SEGURIDAD

- v **PRINCIPIOS DEL DISEÑO PARA LA SEGURIDAD.**
- v EL DISEÑO DEL SISTEMA DEBE SER PUBLICO:
 - v PENSAR QUE EL INTRUSO NO CONOCERA LA FORMA DE FUNCIONAMIENTO DEL SISTEMA ES UN ENGAÑO.
- v EL ESTADO PREDEFINIDO DEBE SER EL DE NO ACCESO:
 - v LOS ERRORES EN DONDE SE NIEGA EL ACCESO VALIDO SE REPORTAN MAS RAPIDO QUE LOS ERRORES EN DONDE SE PERMITE EL ACCESO NO AUTORIZADO.
- v VERIFICAR LA AUTORIZACION ACTUAL:
 - v EL SISTEMA NO DEBE:
 - Φ VERIFICAR EL PERMISO.
 - Φ DETERMINAR QUE EL ACCESO ESTA PERMITIDO.
 - Φ ABANDONAR ESTA INFORMACION PARA SU USO POSTERIOR.
 - v EL SISTEMA TAMPOCO DEBE:
 - Φ VERIFICAR EL PERMISO AL ABRIR UN ARCHIVO Y NO DESPUES DE ABRIRLO:
 - UN ACCESO HABILITADO PERMANECERIA COMO VALIDO AUNQUE HAYA CAMBIADO LA PROTECCION DEL ARCHIVO.
- v DAR A CADA PROCESO EL MINIMO PRIVILEGIO POSIBLE:
 - v IMPLICA UN ESQUEMA DE “PROTECCION DE GRANO FINO”.
- v EL MECANISMO DE PROTECCION DEBE SER SIMPLE, UNIFORME E INTEGRADO HASTA LAS CAPAS MAS BAJAS DEL SISTEMA:
 - v DOTAR DE SEGURIDAD A UN SISTEMA INSEGURO ES CASI IMPOSIBLE.
 - v LA SEGURIDAD NO ES UNA CARACTERISTICA QUE SE PUEDA AÑADIR FACILMENTE.
- v EL ESQUEMA DE SEGURIDAD DEBE SER SICOLOGICAMENTE ACEPTABLE:
 - v LOS USUARIOS NO DEBEN SENTIR QUE LA PROTECCION DE SUS ARCHIVOS LES IMPLICA DEMASIADO TRABAJO:
 - Φ PODRIAN DEJAR DE PROTEGER SUS ARCHIVOS.
 - Φ SE QUEJARIAN EN CASO DE PROBLEMAS.
 - Φ NO ACEPTARIAN FACILMENTE SU PROPIA CULPA.

SEGURIDAD

- v **AUTENTIFICACION DEL USUARIO.**
- v MUCHOS ESQUEMAS DE PROTECCION SE BASAN EN LA HIPOTESIS DE QUE EL SISTEMA CONOCE LA IDENTIDAD DE CADA USUARIO.
- v LA IDENTIFICACION DE LOS USUARIOS SE CONOCE COMO LA **AUTENTIFICACION DE LOS USUARIOS.**
- v MUCHOS METODOS DE AUTENTIFICACION SE BASAN EN:
 - v LA IDENTIFICACION DE ALGO CONOCIDO POR EL USUARIO.
 - v ALGO QUE POSEE EL USUARIO.
 - v ALGO QUE ES EL USUARIO.
- v **CONTRASEÑAS.**
- v SON LA FORMA DE AUTENTIFICACION MAS UTILIZADA.
- v SON DE FACIL COMPRENSION E IMPLEMENTACION.
- v DEBEN ALMACENARSE CIFRADAS (ENCRIPTADAS).
- v SE DEBEN PREVER INTENTOS DE PENETRACION CONSISTENTES EN PRUEBAS DE COMBINACIONES DE NOMBRES Y CONTRASEÑAS.
- v SI LAS CONTRASEÑAS FUERAN DE 7 CARACTERES ELEGIDOS AL AZAR DE LOS 95 CARACTERES ASCII QUE SE PUEDEN IMPRIMIR:
 - v EL ESPACIO DE BUSQUEDA SERIA DE 95^7 , ALREDEDOR DE 7×10^{13} .
 - v A 1.000 CIFRAMIENTOS POR SEGUNDO TOMARIA 2.000 AÑOS CONSTRUIR LA LISTA A VERIFICAR CONTRA EL ARCHIVO DE CONTRASEÑAS.
- v UNA MEJORA AL ESQUEMA DE CONTRASEÑAS CONSISTE EN:
 - v ASOCIAR UN NUMERO ALEATORIO DE “n” BITS A CADA CONTRASEÑA.
 - v EL NUMERO ALEATORIO SE MODIFICA AL CAMBIAR LA CONTRASEÑA.
 - v EL NUMERO SE GUARDA EN EL ARCHIVO DE CONTRASEÑAS EN FORMA NO CIFRADA.
 - v SE CONCATENAN LA CONTRASEÑA Y EL NUMERO ALEATORIO Y SE CIFRAN JUNTOS.
 - v EL RESULTADO CIFRADO SE ALMACENA EN EL ARCHIVO DE CONTRASEÑAS.
 - v SE AUMENTA POR 2ⁿ EL ESPECTRO DE BUSQUEDA:

SEGURIDAD

- v UNA PROTECCION ADICIONAL CONSISTE EN HACER ILEGIBLE EL ARCHIVO DE CONTRASEÑAS ENCRYPTADAS.
- v OTRA PROTECCION ADICIONAL CONSISTE EN QUE EL SISTEMA SUGIERA A LOS USUARIOS CONTRASEÑAS GENERADAS SEGUN CIERTOS CRITERIOS:
 - u SE EVITA QUE EL USUARIO ELIJA CONTRASEÑAS MUY SENCILLAS.
- v TAMBIEN ES CONVENIENTE QUE EL SISTEMA OBLIGUE AL USUARIO A CAMBIAR SUS CONTRASEÑAS CON REGULARIDAD:
 - u SE PUEDE LLEGAR A LA **CONTRASEÑA DE UNA SOLA VEZ.**
- v UNA VARIANTE DE LA IDEA DE CONTRASEÑA ES SOLICITAR AL USUARIO RESPUESTAS SOBRE INFORMACION DE CONTEXTO QUE DEBE CONOCER.
- v OTRA VARIANTE ES LA DE **RETO-RESPUESTA:**
 - u SE ACUERDAN CON EL USUARIO ALGORITMOS (POR EJEMPLO FORMULAS MATEMATICAS) QUE SE UTILIZARAN SEGUN EL DIA Y/O LA HORA.
 - u CUANDO EL USUARIO SE CONECTA:
 - ⊕ EL SISTEMA SUMINISTRA UN ARGUMENTO.
 - ⊕ EL USUARIO DEBE RESPONDER CON EL RESULTADO CORRESPONDIENTE AL ALGORITMO VIGENTE ESE DIA A ESA HORA.
- v **IDENTIFICACION FISICA.**
- v UNA POSIBILIDAD ES LA VERIFICACION DE SI EL USUARIO TIENE CIERTO ELEMENTO (GENERALMENTE UNA TARJETA PLASTICA CON UNA BANDA MAGNETICA):
 - u GENERALMENTE SE COMBINA CON UNA CONTRASEÑA.
- v OTRO ASPECTO CONSISTE EN LA MEDICION DE CARACTERISTICAS FISICAS DIFICILES DE REPRODUCIR:
 - u HUELLAS DIGITALES O VOCALES, FIRMAS, LONGITUD DE LOS DEDOS DE LAS MANOS.
- v **MEDIDAS PREVENTIVAS.**
- v LIMITAR LOS INTENTOS DE ACCESO FALLIDOS Y REGISTRARLOS.
- v REGISTRAR TODOS LOS ACCESOS.
- v TENDER TRAMPAS PARA TRAMPAS DE LOS INTROS.

MECANISMOS DE PROTECCION

- v **DOMINIOS DE PROTECCION.**
- v MUCHOS OBJETOS DEL SISTEMA NECESITAN PROTECCION:
 - u CPU, SEGMENTOS DE MEMORIA, UNIDADES DE DISCO, TERMINALES, IMPRESORAS, PROCESOS, ARCHIVOS, BASES DE DATOS, ETC.
- v CADA OBJETO SE REFERENCIA POR UN NOMBRE Y TIENE HABILITADAS UN CONJUNTO DE OPERACIONES SOBRE EL.
- v UN **DOMINIO** ES UN CONJUNTO DE PAREJAS (OBJETO, DERECHOS):
 - u CADA PAREJA DETERMINA:
 - Φ UN OBJETO.
 - Φ UN SUBCONJUNTO DE LAS OPERACIONES QUE SE PUEDEN LLEVAR A CABO EN EL.
- v UN **DERECHO** ES EL PERMISO PARA REALIZAR ALGUNA DE LAS OPERACIONES.
- v ES POSIBLE QUE UN OBJETO SE ENCUENTRE EN VARIOS DOMINIOS CON “DISTINTOS” DERECHOS EN CADA DOMINIO.
- v UN PROCESO SE EJECUTA EN ALGUNO DE LOS DOMINIOS DE PROTECCION:
 - u EXISTE UNA COLECCION DE OBJETOS A LOS QUE PUEDE TENER ACCESO.
 - u CADA OBJETO TIENE CIERTO CONJUNTO DE DERECHOS.
- v LOS PROCESOS PUEDEN ALTERNAR ENTRE LOS DOMINIOS DURANTE LA EJECUCION.
- v UNA LLAMADA AL S. O. PROVOCA UNA ALTERNANCIA DE DOMINIO.
- v EN ALGUNOS S. O. LOS **DOMINIOS** SE LLAMAN **ANILLOS**.
- v UNA FORMA EN LA QUE EL S. O. LLEVA UN REGISTRO DE LOS OBJETOS QUE PERTENECEN A CADA DOMINIO ES MEDIANTE UNA MATRIZ:
 - u LOS RENGLONES SON LOS DOMINIOS.
 - u LAS COLUMNAS SON LOS OBJETOS.
 - u CADA ELEMENTO DE LA MATRIZ CONTIENE LOS DERECHOS CORRESPONDIENTES AL OBJETO EN ESE DOMINIO:
 - Φ EJ.: LEER, ESCRIBIR, EJECUTAR.

MECANISMOS DE PROTECCION

- v **LISTAS PARA CONTROL DE ACCESO.**
- v LAS “MATRICES DE PROTECCION” SON MUY GRANDES Y CON MUCHOS LUGARES VACIOS:
 - u DESPERDICIAN ESPACIO DE ALMACENAMIENTO.
 - u EXISTEN METODOS PRACTICOS QUE ALMACENAN SOLO LOS ELEMENTOS NO VACIOS POR FILAS O POR COLUMNAS.
- v **LA LISTA DE CONTROL DE ACCESO (ACL: ACCESS CONTROL LIST):**
 - u ASOCIA A CADA OBJETO UNA LISTA ORDENADA CON:
 - Φ TODOS LOS DOMINIOS QUE PUEDEN TENER ACCESO AL OBJETO.
 - Φ LA FORMA DE DICHO ACCESO (EJ: LECTURA (R), GRABACION (W), EJECUCION (X)).
- v UNA FORMA DE IMPLEMENTAR LAS ACL CONSISTE EN:
 - u ASIGNAR TRES BITS (R, W, X) PARA CADA ARCHIVO, PARA:
 - Φ EL PROPIETARIO, EL GRUPO DEL PROPIETARIO Y LOS DEMAS USUARIOS.
 - u PERMITIR QUE EL PROPIETARIO DE CADA OBJETO PUEDA MODIFICAR SU ACL EN CUALQUIER MOMENTO:
 - Φ PERMITE PROHIBIR ACCESOS ANTES PERMITIDOS.
- v **POSIBILIDADES.**
- v LA MATRIZ DE PROTECCION TAMBIEN PUEDE DIVIDIRSE POR RENGLONES:
 - u SE LE ASOCIA A CADA PROCESO UNA **LISTA DE OBJETOS** A LOS CUALES PUEDE TENER ACCESO.
 - u SE LE INDICAN LAS OPERACIONES PERMITIDAS EN CADA UNO.
 - u ESTO DEFINE SU **DOMINIO**.
- v LA LISTA DE OBJETOS SE DENOMINA **LISTA DE POSIBILIDADES** Y LOS ELEMENTOS INDIVIDUALES SE LLAMAN **POSIBILIDADES**.

MECANISMOS DE PROTECCION

- v CADA POSIBILIDAD TIENE:
 - u UN CAMPO *TIPO*:
 - Φ INDICA EL TIPO DEL OBJETO.
 - u UN CAMPO *DERECHOS*:
 - Φ MAPA DE BITS QUE INDICA LAS OPERACIONES BASICAS PERMITIDAS EN ESTE TIPO DE OBJETO.
 - u UN CAMPO *OBJETO*:
 - Φ APUNTADOR AL PROPIO OBJETO (POR EJ.: SU N° DE NODO-I).
- v LAS LISTAS DE POSIBILIDADES SON A SU VEZ OBJETOS Y SE LES PUEDE APUNTAR DESDE OTRAS LISTAS DE POSIBILIDADES:
 - u FACILITA LA EXISTENCIA DE SUBDOMINIOS COMPARTIDOS.
- v LAS **LISTAS DE POSIBILIDADES** O **LISTAS-C** DEBEN SER PROTEGIDAS DEL MANEJO INDEBIDO POR PARTE DEL USUARIO.
- v LOS PRINCIPALES **METODOS DE PROTECCION** SON:
 - u **ARQUITECTURA MARCADA**:
 - Φ NECESITA UN DISEÑO DE HARDWARE EN EL QUE CADA PALABRA DE MEMORIA TIENE UN BIT ADICIONAL:
 - INDICA SI LA PALABRA CONTIENE UNA POSIBILIDAD O NO.
 - SOLO PUEDE SER MODIFICADO POR EL S. O.
 - u **LISTA DE POSIBILIDADES DENTRO DEL S. O.:**
 - Φ LOS PROCESOS HACEN REFERENCIA A LAS POSIBILIDADES MEDIANTE SU NUMERO.
 - u **LISTA DE POSIBILIDADES CIFRADA DENTRO DEL ESPACIO DEL USUARIO:**
 - Φ CADA POSIBILIDAD ESTA CIFRADA CON UNA CLAVE SECRETA DESCONOCIDA POR EL USUARIO.
 - Φ MUY ADECUADO PARA SISTEMAS DISTRIBUIDOS.

MECANISMOS DE PROTECCION

- v GENERALMENTE LAS POSIBILIDADES TIENEN **DERECHOS GENERICOS** APLICABLES A TODOS LOS OBJETOS, POR EJ.:
 - u COPIAR POSIBILIDAD:
 - Φ CREAR UNA NUEVA POSIBILIDAD PARA EL MISMO OBJETO.
 - u COPIAR OBJETO:
 - Φ CREAR UN DUPLICADO DEL OBJETO CON UNA NUEVA POSIBILIDAD.
 - u ELIMINAR POSIBILIDAD:
 - Φ ELIMINAR UN DATO DENTRO DE LA LISTA-C SIN AFECTAR AL OBJETO.
 - u DESTRUIR OBJETO:
 - Φ ELIMINAR EN FORMA PERMANENTE UN OBJETO Y UNA POSIBILIDAD.
- v MUCHOS SISTEMAS CON POSIBILIDADES SE ORGANIZAN COMO UNA COLECCION DE MODULOS CON **MODULOS ADMINISTRADORES DE TIPOS** PARA CADA TIPO DE OBJETO:
 - u ES ESENCIAL QUE EL MODULO ADMINISTRADOR DE TIPOS PUEDA HACER MAS COSAS CON LA POSIBILIDAD QUE UN PROCESO ORDINARIO.
- v SE UTILIZA LA TECNICA DE **AMPLIFICACION DE DERECHOS**:
 - u LOS ADMINISTRADORES DE TIPO OBTIENEN UNA PLANTILLA DE DERECHOS QUE LES DA MAS DERECHOS SOBRE UN OBJETO DE LOS QUE PERMITIA LA PROPIA LISTA DE POSIBILIDADES.
- v **MODELOS DE PROTECCION.**
- v LAS MATRICES DE PROTECCION NO SON ESTATICAS SINO DINAMICAS.
- v SE PUEDEN IDENTIFICAR SEIS OPERACIONES PRIMITIVAS EN LA MATRIZ DE PROTECCION:
 - u CREAR OBJETO.
 - u ELIMINAR OBJETO.
 - u CREAR DOMINIO.
 - u ELIMINAR DOMINIO.
 - u INSERTAR DERECHO.
 - u ELIMINAR DERECHO.

MECANISMOS DE PROTECCION

- v LAS PRIMITIVAS SE PUEDEN COMBINAR EN **COMANDOS DE PROTECCION**:
 - u PUEDEN SER EJECUTADOS POR LOS PROGRAMAS DEL USUARIO PARA MODIFICAR LA MATRIZ DE PROTECCION.
- v EN CADA MOMENTO, LA MATRIZ DE PROTECCION DETERMINA LO QUE PUEDE HACER UN PROCESO EN CUALQUIER MOMENTO:
 - u NO DETERMINA LO QUE NO ESTA AUTORIZADO A REALIZAR.
- v LA MATRIZ ES IMPUESTA POR EL SISTEMA.
- v LA AUTORIZACION TIENE QUE VER CON LA POLITICA DE ADMINISTRACION.
- v **CONTROL DE ACCESO POR CLASES DE USUARIOS.**
- v UNA MATRIZ DE CONTROL DE ACCESO PUEDE LLEGAR A SER TAN GRANDE QUE RESULTE IMPRACTICO MANTENERLA.
- v UNA TECNICA QUE REQUIERE MENOS ESPACIO ES CONTROLAR EL ACCESO A VARIAS *CLASES DE USUARIOS*.
- v EJ. DE ESQUEMA DE CLASIFICACION:
 - u *PROPIETARIO*:
 - Φ SUELE SER EL USUARIO QUE CREO EL ARCHIVO.
 - u *USUARIO ESPECIFICADO*:
 - Φ EL PROPIETARIO ESPECIFICA QUIEN MAS PUEDE USAR EL ARCHIVO.
 - u *GRUPO O PROYECTO*:
 - Φ LOS DIFERENTES MIEMBROS DE UN GRUPO DE TRABAJO SOBRE UN PROYECTO, ACCEDEN A LOS DIFERENTES ARCHIVOS RELACIONADOS CON EL PROYECTO.
 - u *PUBLICO*:
 - Φ UN ARCHIVO PUBLICO PUEDE SER ACCEDIDO POR CUALQUIER USUARIO DE LA COMPUTADORA.
 - Φ GENERALMENTE PERMITE LEER O EJECUTAR PERO NO ESCRIBIR SOBRE EL ARCHIVO.

RESPALDO Y RECUPERACION

- v LA DESTRUCCION DE LA INFORMACION, YA SEA ACCIDENTAL O INTENCIONAL, ES UNA REALIDAD Y TIENE DISTINTAS CAUSAS:
 - u FALLAS DE HARDWARE Y DE SOFTWARE.
 - u FENOMENOS METEOROLOGICOS ATMOSFERICOS.
 - u FALLAS EN EL SUMINISTRO DE ENERGIA.
 - u INCENDIOS E INUNDACIONES.
 - u ROBOS, VANDALISMO (INCLUSO TERRORISMO).
 - u ETC.
- v ESTA POSIBLE DESTRUCCION DE LA INFORMACION DEBE SER TENIDA EN CUENTA POR:
 - u LOS SISTEMAS OPERATIVOS EN GENERAL.
 - u LOS SISTEMAS DE ARCHIVOS EN PARTICULAR.
- v UNA TECNICA MUY USADA PARA ASEGURAR LA DISPONIBILIDAD DE LOS DATOS ES REALIZAR *RESPALDOS PERIODICOS*:
 - u HACER CON REGULARIDAD UNA O MAS COPIAS DE LOS ARCHIVOS Y COLOCARLAS EN LUGAR SEGURO.
 - u TODAS LAS ACTUALIZACIONES REALIZADAS LUEGO DEL ULTIMO RESPALDO PUEDEN PERDERSE.
- v OTRA TECNICA ES *PASAR TODAS LAS TRANSACCIONES A UN ARCHIVO*, COPIANDOLAS EN OTRO DISCO:
 - u GENERA UNA *REDUNDANCIA* QUE PUEDE SER COSTOSA.
 - u EN CASO DE FALLAS EN EL DISCO PRINCIPAL, PUEDE RECONSTRUIRSE TODO EL TRABAJO PERDIDO SI EL *DISCO DE RESERVA* NO SE DAÑO TAMBIEN.
- v TAMBIEN EXISTE LA POSIBILIDAD DEL *RESPALDO INCREMENTAL*:
 - u DURANTE UNA SESION DE TRABAJO LOS ARCHIVOS MODIFICADOS QUEDAN MARCADOS.
 - u CUANDO UN USUARIO SE RETIRA DEL SISTEMA (DEJA DE TRABAJAR):
 - Φ UN PROCESO DEL SISTEMA EFECTUA EL RESPALDO DE LOS ARCHIVOS MARCADOS.
- v SE DEBE TENER PRESENTE QUE *ES MUY DIFICIL GARANTIZAR UNA*

ENTRADA / SALIDA

- v INTRODUCCION
- v PRINCIPIOS DEL HARDWARE DE E / S
- v PRINCIPIOS DEL SOFTWARE DE E / S
- v DISCOS
- v HARDWARE PARA DISCOS
- v OPERACION DE ALMACENAMIENTO DE DISCO DE CABEZA MOVIL
- v ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO
- v PORQUE ES NECESARIA LA PLANIFICACION DE DISCOS
- v CARACTERISTICAS DESEABLES DE LAS POLITICAS DE PLANIFICACION DE DISCOS
- v OPTIMIZACION DE LA BUSQUEDA EN DISCOS
- v OPTIMIZACION ROTACIONAL EN DISCOS
- v CONSIDERACIONES DE LOS DISCOS SOBRE LOS SISTEMAS
- v MANEJO DE ERRORES EN DISCOS
- v OCULTAMIENTO DE UNA PISTA A LA VEZ EN DISCOS
- v DISCOS EN RAM
- v RELOJES
- v TERMINALES

INTRODUCCION

- v UNA DE LAS FUNCIONES PRINCIPALES DE UN S. O. ES EL CONTROL DE TODOS LOS DISPOSITIVOS DE E / S DE LA COMPUTADORA.
- v LAS PRINCIPALES FUNCIONES RELACIONADAS SON:
 - u ENVIAR COMANDOS A LOS DISPOSITIVOS.
 - u DETECTAR LAS INTERRUPCIONES.
 - u CONTROLAR LOS ERRORES.
 - u PROPORCIONAR UNA INTERFAZ ENTRE LOS DISPOSITIVOS Y EL RESTO DEL SISTEMA:
 - Φ DEBE SER SENCILLA Y FACIL DE USAR.
 - Φ DEBE SER LA MISMA (PREFERENTEMENTE) PARA TODOS LOS DISPOSITIVOS (INDEPENDENCIA DEL DISPOSITIVO).
- v EL CODIGO DE E / S REPRESENTA UNA FRACCION SIGNIFICATIVA DEL S. O.
- v EL USO INAPROPIADO DE LOS DISPOSITIVOS DE E / S FRECUENTEMENTE GENERA INEFICIENCIAS DEL SISTEMA:
 - u AFECTA LA PERFORMANCE GLOBAL.

PRINCIPIOS DEL HARDWARE DE E / S

- v EL ENFOQUE QUE SE CONSIDERARA TIENE QUE VER CON LA INTERFAZ QUE DESDE EL HARDWARE SE PRESENTA AL SOFTWARE:
 - u COMANDOS QUE ACEPTA EL HARDWARE, FUNCIONES QUE REALIZA Y ERRORES QUE PUEDE INFORMAR.
- v **DISPOSITIVOS DE E / S.**
- v SE PUEDEN CLASIFICAR EN DOS GRANDES CATEGORIAS:
 - u DISPOSITIVOS DE BLOQUE.
 - u DISPOSITIVOS DE CARACTER.
- v LAS PRINCIPALES CARACTERISTICAS DE LOS D. DE BLOQUE SON:
 - u LA INFORMACION SE ALMACENA EN BLOQUES DE TAMAÑO FIJO.
 - u CADA BLOQUE TIENE SU PROPIA DIRECCION.
 - u LOS TAMAÑOS MAS COMUNES DE LOS BLOQUES VAN DESDE LOS 128 BYTES HASTA LOS 1.024 BYTES.
 - u SE PUEDE LEER O ESCRIBIR EN UN BLOQUE DE FORMA INDEPENDIENTE DE LOS DEMAS, EN CUALQUIER MOMENTO.
 - u EJEMPLO TIPICO DE DISPOSITIVOS DE BLOQUE: DISCOS.
- v LAS PRINCIPALES CARACTERISTICAS DE LOS D. DE CARACTER SON:
 - u LA INFORMACION SE TRANSFIERE COMO UN FLUJO DE CARACTERES, SIN SUJETARSE A UNA ESTRUCTURA DE BLOQUES.
 - u NO SE PUEDEN UTILIZAR DIRECCIONES.
 - u NO TIENEN UNA OPERACION DE BUSQUEDA.
 - u EJEMPLOS TIPICO DE DISPOSITIVOS DE CARACTER: IMPRESORAS DE LINEA, TERMINALES, INTERFACES DE UNA RED, RATONES, ETC.
- v ALGUNOS DISPOSITIVOS NO SE AJUSTAN A ESTE ESQUEMA DE CLASIFICACION:
 - u LOS RELOJES NO TIENEN DIRECCIONES POR MEDIO DE BLOQUES Y NO GENERAN O ACEPTAN FLUJOS DE CARACTERES.
- v EL SISTEMA DE ARCHIVOS SOLO TRABAJA CON DISPOSITIVOS DE BLOQUE ABSTRACTOS:
 - u ENCARGA LA PARTE DEPENDIENTE DEL DISPOSITIVO A UN SOFTWARE DE MENOR NIVEL:
 - ❖ **SOFTWARE MANEJADOR DEL DISPOSITIVO.**

PRINCIPIOS DEL HARDWARE DE E / S

- v **CONTROLADORES DE DISPOSITIVOS.**
- v LAS UNIDADES DE E / S GENERALMENTE CONSTAN DE:
 - u UN COMPONENTE MECANICO.
 - u UN COMPONENTE ELECTRONICO:
 - Φ **CONTROLADOR DEL DISPOSITIVO O ADAPTADOR.**
- v MUCHOS CONTROLADORES PUEDEN MANEJAR MAS DE UN DISPOSITIVO.
- v EL S. O. GENERALMENTE TRABAJA CON EL CONTROLADOR Y NO CON EL DISPOSITIVO.
- v LOS MODELOS MAS FRECUENTES DE COMUNICACION ENTRE LA CPU Y LOS CONTROLADORES SON:
 - u PARA LA MAYORIA DE LAS MICRO Y MINI COMPUTADORAS:
 - Φ **MODELO DE BUS DEL SISTEMA.**
 - u PARA LA MAYORIA DE LOS MAINFRAMES:
 - Φ **MODELO DE VARIOS BUSES Y COMPUTADORAS ESPECIALIZADAS EN E / S LLAMADAS CANALES DE E / S.**
- v LA INTERFAZ ENTRE EL CONTROLADOR Y EL DISPOSITIVO ES CON FRECUENCIA DE MUY BAJO NIVEL:
 - u LA COMUNICACION ES MEDIANTE UN FLUJO DE BITS EN SERIE QUE:
 - Φ **COMIENZA CON UN PREAMBULO.**
 - Φ **SIGUE CON UNA SERIE DE BITS (DE UN SECTOR DE DISCO, POR EJ.).**
 - Φ **CONCLUYE CON UNA SUMA PARA VERIFICACION O UN CODIGO CORRECTOR DE ERRORES.**
 - u **EL PREAMBULO:**
 - Φ **SE ESCRIBE AL DAR FORMATO AL DISCO.**
 - Φ **CONTIENE EL N° DE CILINDRO Y SECTOR, EL TAMAÑO DE SECTOR Y OTROS DATOS SIMILARES.**
- v EL CONTROLADOR DEBE:
 - u **CONVERTIR EL FLUJO DE BITS EN SERIE EN UN BLOQUE DE BYTES.**
 - u **EFFECTUAR CUALQUIER CORRECCION DE ERRORES NECESARIA.**
 - u **COPIAR EL BLOQUE EN LA MEMORIA PRINCIPAL.**

PRINCIPIOS DEL HARDWARE DE E / S

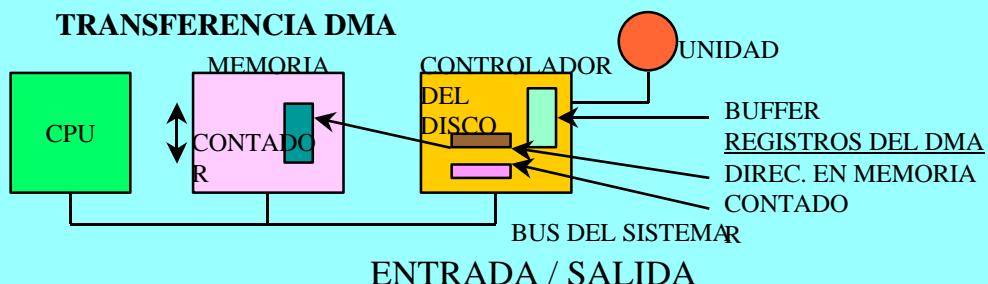
- v CADA CONTROLADOR POSEE REGISTROS QUE UTILIZA PARA COMUNICARSE CON LA CPU:
 - PUEDEN SER PARTE DEL ESPACIO NORMAL DE DIRECCIONES DE LA MEMORIA: **E / S MAPEADA A MEMORIA.**
 - PUEDEN UTILIZAR UN ESPACIO DE DIRECCIONES ESPECIAL PARA LA E / S, ASIGNANDO A CADA CONTROLADOR UNA PARTE DE EL.
- v EL S. O. REALIZA LA E / S AL ESCRIBIR COMANDOS EN LOS REGISTROS DE LOS CONTROLADORES:
 - LOS PARAMETROS DE LOS COMANDOS TAMBIEN SE CARGAN EN LOS REGISTROS DE LOS CONTROLADORES.
- v AL ACEPTAR EL COMANDO, LA CPU PUEDE DEJAR AL CONTROLADOR Y DEDICARSE A OTRO TRABAJO.
- v AL TERMINAR EL COMANDO, EL CONTROLADOR PROVOCA UNA INTERRUPCION PARA PERMITIR QUE EL S. O.:
 - OBTENGA EL CONTROL DE LA CPU.
 - VERIFIQUE LOS RESULTADOS DE LA OPERACION.
- v LA CPU OBTIENE LOS RESULTADOS Y EL ESTADO DEL DISPOSITIVO AL LEER UNO O MAS BYTES DE INFORMACION DE LOS REGISTROS DEL CONTROLADOR.
- v EJEMPLOS DE CONTROLADORES, SUS DIRECCIONES DE E / S Y SUS VECTORES DE INTERRUPCION EN LA PC IBM:

<u>CONTROLADOR DE E / S</u>	<u>DIRECCION DE E / S</u>	<u>VECTOR DE INTER.</u>
▫ RELOJ	040 - 043	8
▫ TECLADO	060 - 063	9
▫ DISCO DURO	320 - 32F	13
▫ IMPRESORA	378 - 37F	15
▫ DISCO FLEXIBLE	3F0 - 3F7	14
▫ RS232 PRIMARIO	3F8 - 3FF	12
▫ RS232 SECUNDARIO	2F8 - 2FF	11

PRINCIPIOS DEL HARDWARE DE E / S

- v **ACCESO DIRECTO A MEMORIA (DMA).**
- v MUCHOS CONTROLADORES, ESPECIALMENTE LOS CORRESPONDIENTES A DISPOSITIVOS DE BLOQUE, PERMITEN EL DMA.
- v SI SE LEE EL DISCO SIN DMA:
 - υ EL CONTROLADOR LEE EN SERIE EL BLOQUE (UNO O MAS SECTORES) DE LA UNIDAD:
 - Φ LA LECTURA ES BIT POR BIT.
 - Φ LOS BITS DEL BLOQUE SE GRABAN EN EL BUFFER INTERNO DEL CONTROLADOR.
 - υ SE CALCULA LA SUMA DE VERIFICACION PARA CORROBORAR QUE NO EXISTEN ERRORES DE LECTURA.
 - υ EL CONTROLADOR PROVOCA UNA INTERRUPCION.
 - υ EL S. O. LEE EL BLOQUE DEL DISCO POR MEDIO DEL BUFFER DEL CONTROLADOR:
 - Φ LA LECTURA ES POR BYTE O PALABRA A LA VEZ.
 - Φ EN CADA ITERACION DE ESTE CICLO SE LEE UN BYTE O UNA PALABRA DEL REGISTRO DEL CONTROLADOR Y SE ALMACENA EN MEMORIA.
 - υ SE DESPERDICIA TIEMPO DE LA CPU.
- v DMA SE IDEO PARA LIBERAR A LA CPU DE ESTE TRABAJO DE BAJO NIVEL.
- v LA CPU LE PROPORCIONA AL CONTROLADOR:
 - υ LA DIRECCION DEL BLOQUE EN EL DISCO.
 - υ LA DIRECCION EN MEMORIA ADONDE DEBE IR EL BLOQUE.
 - υ EL N° DE BYTES POR TRANSFERIR.

UN CONTROLADOR REALIZA COMPLETAMENTE UNA TRANSFERENCIA DMA



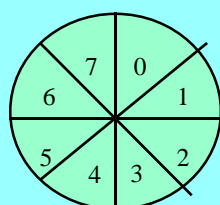
PRINCIPIOS DEL HARDWARE DE E / S

- v LUEGO DE QUE EL CONTROLADOR LEYO TODO EL BLOQUE DEL DISPOSITIVO A SU BUFFER Y DE QUE CORROBORO LA SUMA DE VERIFICACION:
 - u COPIA EL PRIMER BYTE O PALABRA A LA MEMORIA PRINCIPAL.
 - u LO HACE EN LA DIRECCION ESPECIFICADA POR MEDIO DE LA DIRECCION DE MEMORIA DE DMA.
 - u INCREMENTA LA DIRECCION DMA Y DECREMENTA EL CONTADOR DMA EN EL N° DE BYTES QUE ACABA DE TRANSFERIR.
 - u SE REPITE ESTE PROCESO HASTA QUE EL CONTADOR SE ANULA:
 - Φ EL CONTROLADOR PROVOCA UNA INTERRUPCION.
 - u AL INICIAR SU EJECUCION EL S. O. LUEGO DE LA INTERRUPCION PROVOCADA:
 - Φ NO DEBE COPIAR EL BLOQUE EN LA MEMORIA:
 - YA SE ENCUENTRA AHI.
- v EL CONTROLADOR NECESITA UN BUFFER INTERNO PORQUE UNA VEZ INICIADA UNA TRANSFERENCIA DEL DISCO:
 - u LOS BITS SIGUEN LLEGANDO DEL DISCO CONSTANTEMENTE.
 - u NO INTERESA SI EL CONTROLADOR ESTA LISTO O NO PARA RECIBIRLOS.
 - u SI EL CONTROLADOR INTENTARA ESCRIBIR LOS DATOS EN LA MEMORIA DIRECTAMENTE:
 - Φ TENDRIA QUE RECURRIR AL BUS DEL SISTEMA PARA C / U DE LAS PALABRAS (O BYTES) TRANSFERIDAS.
 - Φ EL BUS PODRIA ESTAR OCUPADO POR OTRO DISPOSITIVO Y EL CONTROLADOR DEBERIA ESPERAR.
 - Φ SI LA SIGUIENTE PALABRA LLEGARA ANTES DE QUE LA ANTERIOR HUBIERA SIDO ALMACENADA:
 - EL CONTROLADOR LA TENDRIA QUE ALMACENAR EN ALGUNA PARTE.
- v SI EL BLOQUE SE GUARDA EN UN BUFFER INTERNO:
 - u EL BUS NO SE NECESITA SINO HASTA QUE EL DMA COMIENZA.
 - u LA TRANSFERENCIA DMA A LA MEMORIA YA NO ES UN ASPECTO CRITICO DEL TIEMPO.

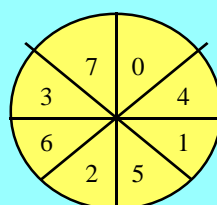
PRINCIPIOS DEL HARDWARE DE E / S

- v LOS CONTROLADORES SIMPLES NO PUEDEN ATENDER LA E / S SIMULTANEA:
 - υ MIENTRAS TRANSFIEREN A LA MEMORIA, EL SECTOR QUE PASA DEBAJO DE LA CABEZA DEL DISCO SE PIERDE:
 - Φ EL BLOQUE SIGUIENTE AL RECIEN LEIDO SE PIERDE.
 - υ LA LECTURA DE UNA PISTA COMPLETA SE HARA EN DOS ROTACIONES COMPLETAS:
 - Φ UNA PARA LOS BLOQUES PARES Y OTRA PARA LOS IMPARES.
 - υ SI EL TIEMPO NECESARIO PARA UNA TRANSFERENCIA DE UN BLOQUE DEL CONTROLADOR A LA MEMORIA POR MEDIO DEL BUS ES MAYOR QUE EL TIEMPO NECESARIO PARA LEER UN BLOQUE DEL DISCO:
 - Φ SERIA NECESARIO LEER UN BLOQUE Y LUEGO SALTAR DOS O MAS BLOQUES.
 - Φ EL SALTO DE BLOQUES:
 - SE EJECUTA PARA DARLE TIEMPO AL CONTROLADOR PARA LA TRANSFERENCIA DE LOS DATOS A LA MEMORIA.
 - SE LLAMA **SEPARACION**.
 - AL FORMATEAR EL DISCO, LOS BLOQUES SE NUMERAN TOMANDO EN CUENTA EL **FACTOR DE SEPARACION**.
 - ESTO PERMITE AL S. O.:
 - LEER LOS BLOQUES CON NUMERACION CONSECUTIVA.
 - CONSERVAR LA MAXIMA VELOCIDAD POSIBLE DEL HARDWARE.

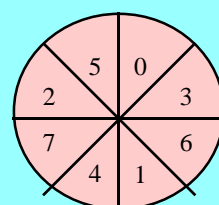
SIN SEPARACION



SEPARACION SIMPLE



SEPARACION DOBLE



ENTRADA / SALIDA

PRINCIPIOS DEL SOFTWARE DE E / S

- v LA IDEA BASICA ES ORGANIZAR EL SOFTWARE COMO UNA SERIE DE CAPAS DONDE:
 - u LAS CAPAS INFERIORES SE ENCARGUEN DE OCULTAR LAS PECULIARIDADES DEL HARDWARE A LAS CAPAS SUPERIORES.
 - u LAS CAPAS SUPERIORES DEBEN PRESENTAR UNA INTERFAZ AGRADABLE, LIMPIA Y REGULAR A LOS USUARIOS.
- v **OBJETIVOS DEL SOFTWARE DE E / S.**
- v UN CONCEPTO CLAVE ES LA **INDEPENDENCIA DEL DISPOSITIVO**:
 - u DEBE SER POSIBLE ESCRIBIR PROGRAMAS QUE SE PUEDAN UTILIZAR CON ARCHIVOS EN DISTINTOS DISPOSITIVOS:
 - Φ SIN TENER QUE MODIFICAR LOS PROGRAMAS PARA CADA TIPO DE DISPOSITIVO.
 - u EL PROBLEMA DEBE SER RESUELTO POR EL S. O.
- v EL OBJETIVO DE LOGRAR **NOMBRES UNIFORMES** ESTA MUY RELACIONADO CON EL DE INDEPENDENCIA DEL DISPOSITIVO.
- v TODOS LOS ARCHIVOS Y DISPOSITIVOS ADQUIEREN DIRECCIONES DE LA MISMA FORMA:
 - u MEDIANTE EL NOMBRE DE SU RUTA DE ACCESO.
- v OTRO ASPECTO IMPORTANTE DEL SOFTWARE ES EL MANEJO DE ERRORES DE E / S:
 - u GENERALMENTE LOS ERRORES DEBEN MANEJARSE LO MAS CERCA POSIBLE DEL HARDWARE.
 - u SOLO SI LOS NIVELES INFERIORES NO PUEDEN RESOLVER EL PROBLEMA, SE INFORMA A LOS NIVELES SUPERIORES.
 - u GENERALMENTE LA RECUPERACION SE PUEDE HACER EN UN NIVEL INFERIOR Y DE FORMA TRANSPARENTE.
- v OTRO ASPECTO CLAVE SON LAS **TRANSFERENCIAS SINCRONAS** (POR BLOQUES) **O ASINCRONAS** (CONTROLADA POR INTERRUPTORES):
 - u LA MAYORIA DE LA E / S ES **ASINCRONA**:
 - Φ LA CPU INICIA LA TRANSFERENCIA Y REALIZA OTRAS TAREAS HASTA UNA INTERRUPCION.
 - u LA PROGRAMACION ES MAS FACIL SI LA E / S ES **SINCRONA** (POR BLOQUES):
 - Φ EL PROGRAMA SE SUSPENDE AUTOMATICAMENTE HASTA **ENTRADA / SALIDA** QUE LOS DATOS ESTEN DISPONIBLES EN EL BUFFER.

PRINCIPIOS DEL SOFTWARE DE E / S

- v EL S. O. SE ENCARGA DE HACER QUE OPERACIONES CONTROLADAS POR INTERRUPTORES PAREZCAN DEL TIPO DE BLOQUES PARA EL USUARIO.
- v TAMBIEN EL S. O. DEBE ADMINISTRAR LOS **DISPOSITIVOS COMPARTIDOS** (EJ.: DISCOS) Y LOS **DE USO EXCLUSIVO** (EJ.: IMPRESORAS).
- v GENERALMENTE EL SOFTWARE DE E / S SE ESTRUCTURA EN **CAPAS**:
 - u MANEJADORES DE INTERRUPCIONES.
 - u DIRECTIVAS DE DISPOSITIVOS.
 - u SOFTWARE DE S. O. INDEPENDIENTE DE LOS DISPOSITIVOS.
 - u SOFTWARE A NIVEL USUARIO.
- v **MANEJADORES DE INTERRUPCIONES.**
- v LAS INTERRUPCIONES DEBEN OCULTARSE EN EL S. O.:
 - u CADA PROCESO QUE INICIE UNA OPERACION DE E / S SE BLOQUEA HASTA QUE TERMINA LA E / S Y OCURRA LA INTERRUPCION.
 - u EL PROCEDIMIENTO DE INTERRUPCION REALIZA LO NECESARIO PARA DESBLOQUEAR EL PROCESO QUE LO INICIO.
- v **MANEJADORES DE DISPOSITIVOS.**
- v TODO EL CODIGO QUE DEPENDE DE LOS DISPOSITIVOS APARECE EN LOS MANEJADORES DE DISPOSITIVOS.
- v CADA CONTROLADOR POSEE UNO O MAS REGISTROS DE DISPOSITIVOS:
 - u SE UTILIZAN PARA DARLE LOS COMANDOS.
 - u LOS MANEJADORES DE DISPOSITIVOS PROVEEN ESTOS COMANDOS Y VERIFICAN SU EJECUCION ADECUADA.
- v LA LABOR DE UN MANEJADOR DE DISPOSITIVOS ES LA DE:
 - u ACEPTAR LAS SOLICITUDES ABSTRACTAS QUE LE HACE EL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO.
 - u VERIFICAR LA EJECUCION DE DICHAS SOLICITUDES.
- v SI AL RECIBIR UNA SOLICITUD EL MANEJADOR ESTA OCUPADO CON OTRA SOLICITUD:
 - u AGREGARA LA NUEVA SOLICITUD A UNA COLA DE SOLICITUDES PENDIENTES.

PRINCIPIOS DEL SOFTWARE DE E / S

- v LA SOLICITUD DE E / S, POR EJ. PARA UN DISCO, SE DEBE TRADUCIR DE TERMINOS ABSTRACTOS A TERMINOS CONCRETOS:
 - u EL MANEJADOR DE DISCO DEBE:
 - ⊕ ESTIMAR EL LUGAR DONDE SE ENCUENTRA EN REALIDAD EL BLOQUE SOLICITADO.
 - ⊕ VERIFICAR SI EL MOTOR DE LA UNIDAD FUNCIONA.
 - ⊕ VERIFICAR SI EL BRAZO ESTA COLOCADO EN EL CILINDRO ADECUADO, ETC.
 - ⊕ RESUMIENDO: DEBE DECIDIR:
 - **CUALES SON LAS OPERACIONES NECESARIAS DEL CONTROLADOR Y SU ORDEN.**
 - ⊕ ENVIA LOS COMANDOS AL CONTROLADOR AL ESCRIBIR EN LOS REGISTROS DE DISPOSITIVO DEL MISMO.
 - ⊕ FRECUENTEMENTE EL MANEJADOR DEL DISPOSITIVO SE BLOQUEA HASTA QUE EL CONTROLADOR REALIZA CIERTO TRABAJO:
 - UNA INTERRUPCION LO LIBERA DE ESTE BLOQUEO.
 - ⊕ AL FINALIZAR LA OPERACION DEBE VERIFICAR LOS ERRORES.
 - ⊕ SI TODO ESTA O.K. TRANSFERIRA LOS DATOS AL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO.
 - ⊕ REGRESA INFORMACION DE ESTADO SOBRE LOS ERRORES A QUIEN LO LLAMO.
 - ⊕ INICIA OTRA SOLICITUD PENDIENTE O QUEDA EN ESPERA.
- v **SOFTWARE DE E / S INDEPENDIENTE DEL DISPOSITIVO.**
- v FUNCIONES GENERALMENTE REALIZADAS POR EL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO:
 - u INTERFAZ UNIFORME PARA LOS MANEJADORES DE DISPOSITIVOS.
 - u NOMBRES DE LOS DISPOSITIVOS.
 - u PROTECCION DEL DISPOSITIVO.
 - u PROPORCIONAR UN TAMAÑO DE BLOQUE INDEPENDIENTE DEL DISPOSITIVO.
 - u USO DE BUFFERS.
 - u ASIGNACION DE ESPACIO EN LOS DISPOSITIVOS POR BLOQUES.
 - u ASIGNACION Y LIBERACION DE LOS DISPOSITIVOS DE USO EXCLUSIVO.
- u INFORME DE ERRORES.

PRINCIPIOS DEL SOFTWARE DE E / S

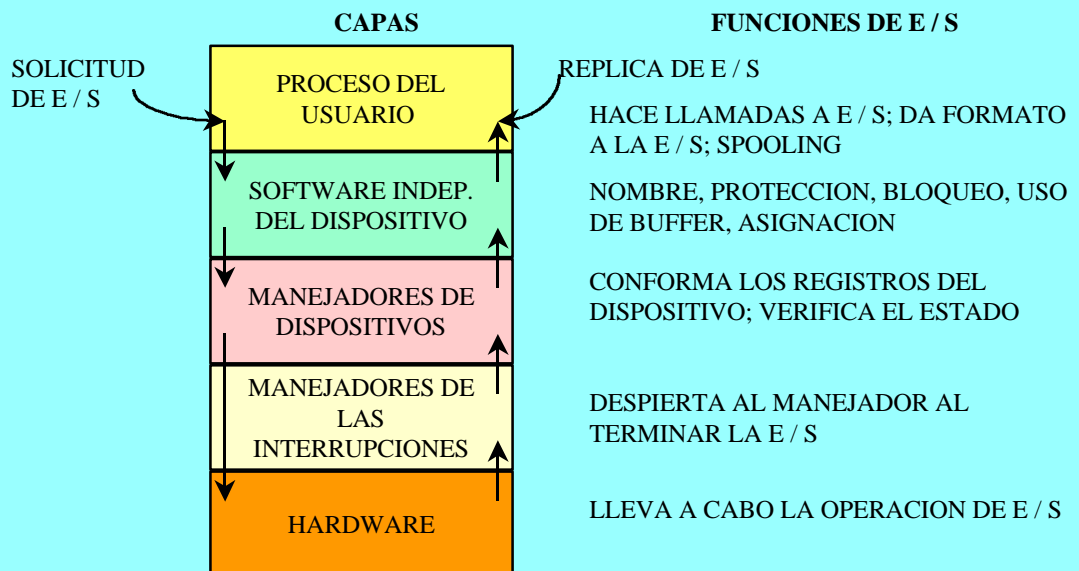
- v LAS FUNCIONES BASICAS DEL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO SON:
 - u EFECTUAR LAS FUNCIONES DE E / S COMUNES A TODOS LOS DISPOSITIVOS.
 - u PROPORCIONAR UNA INTERFAZ UNIFORME DEL SOFTWARE A NIVEL USUARIO.
- v EL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO ASOCIA LOS NOMBRES SIMBOLICOS DE LOS DISPOSITIVOS CON EL NOMBRE ADECUADO.
- v UN NOMBRE DE DISPOSITIVO DETERMINA DE MANERA UNICA EL NODO-I DE UN ARCHIVO ESPECIAL:
 - u ESTE NODO-I CONTIENE EL N° **PRINCIPAL DEL DISPOSITIVO**:
 - Φ SE UTILIZA PARA LOCALIZAR EL MANEJADOR APROPIADO.
 - u EL NODO-I CONTIENE TAMBIEN EL N° **SECUNDARIO DE DISPOSITIVO**:
 - Φ SE TRANSFIERE COMO PARAMETRO AL MANEJADOR PARA DETERMINAR LA UNIDAD POR LEER O ESCRIBIR.
- v EL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO DEBE:
 - u OCULTAR A LOS NIVELES SUPERIORES LOS DIFERENTES TAMAÑOS DE SECTOR DE LOS DISTINTOS DISCOS.
 - u PROPORCIONAR UN TAMAÑO UNIFORME DE LOS BLOQUES:
 - Φ EJ.: CONSIDERAR VARIOS SECTORES FISICOS COMO UN SOLO BLOQUE LOGICO.
- v **SOFTWARE DE E / S EN EL ESPACIO DEL USUARIO.**
- v LA MAYORIA DEL SOFTWARE DE E / S ESTA DENTRO DEL S. O.
- v UNA PEQUEÑA PARTE CONSTA DE BIBLIOTECAS LIGADAS ENTRE SI CON LOS PROGRAMAS DEL USUARIO:
 - u LA BIBLIOTECA ESTANDAR DE E / S CONTIENE VARIOS PROCEDIMIENTOS RELACIONADOS CON E / S Y TODOS SE EJECUTAN COMO PARTE DE LOS PROGRAMAS DEL USUARIO.

PRINCIPIOS DEL SOFTWARE DE E / S

- v OTRA CATEGORIA IMPORTANTE DE SOFTWARE DE E / S A NIVEL USUARIO ES EL **SISTEMA DE SPOOLING**.
- v EL **SPOOLING** ES UNA FORMA DE TRABAJAR CON LOS DISPOSITIVOS DE E / S DE USO EXCLUSIVO EN UN SISTEMA DE MULTIPROGRAMACION:
 - u EL EJEMPLO TIPICO LO CONSTITUYE LA IMPRESORA DE LINEAS.
 - u LOS PROCESOS DE USUARIO NO ABREN EL ARCHIVO CORRESPONDIENTE A LA IMPRESORA.
 - u SE CREA UN **PROCESO ESPECIAL**, LLAMADO **DEMONIO** EN ALGUNOS SISTEMAS.
 - u SE CREA UN **DIRECTORIO DE SPOOLING**.
- v PARA IMPRIMIR UN ARCHIVO:
 - u UN PROCESO GENERA TODO EL ARCHIVO POR IMPRIMIR Y LO COLOCA EN EL DIRECTORIO DE SPOOLING.
 - u EL PROCESO ESPECIAL, UNICO CON PERMISO PARA UTILIZAR EL ARCHIVO ESPECIAL DE LA IMPRESORA, DEBE IMPRIMIR LOS ARCHIVOS EN EL DIRECTORIO.
 - u SE EVITA EL POSIBLE PROBLEMA DE TENER UN PROCESO DE USUARIO QUE MANTENGA UN RECURSO TOMADO LARGO TIEMPO.
- v UN ESQUEMA SIMILAR TAMBIEN ES APLICABLE PARA LA TRANSFERENCIA DE ARCHIVOS ENTRE EQUIPOS CONECTADOS:
 - u UN USUARIO COLOCA UN ARCHIVO EN UN DIRECTORIO DE SPOOLING DE LA RED.
 - u POSTERIORMENTE, EL PROCESO ESPECIAL LO TOMA Y TRANSMITE.
 - u EJ.: SISTEMAS DE CORREO ELECTRONICO.

PRINCIPIOS DEL SOFTWARE DE E / S

CAPAS DEL SISTEMA DE E / S Y LAS PRINCIPALES FUNCIONES DE C/ CAPA



DISCOS

HARDWARE PARA DISCOS

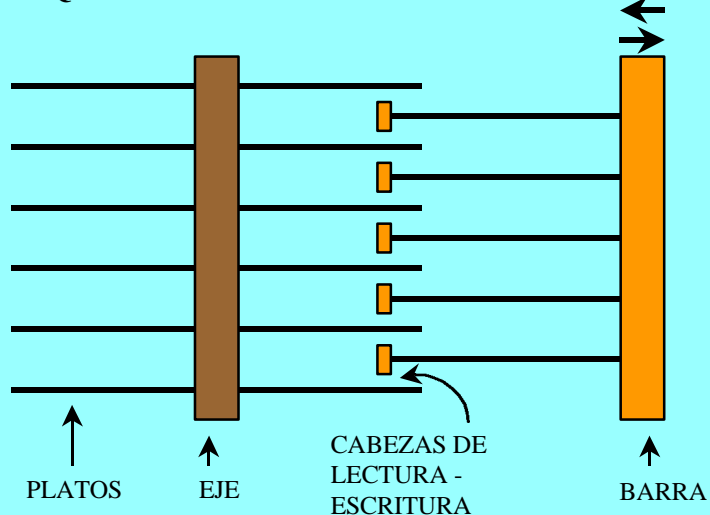
- v **DISCOS.**
- v LAS SIGUIENTES SON LAS PRINCIPALES VENTAJAS CON RESPECTO DEL USO DE LA MEMORIA PRINCIPAL COMO ALMACENAMIENTO:
 - υ MUCHO MAYOR CAPACIDAD DE ESPACIO DE ALMACENAMIENTO.
 - υ MENOR PRECIO POR BIT.
 - υ LA INFORMACION NO SE PIERDE AL APAGAR LA COMPUTADORA.
- v UN USO INAPROPIADO DE LOS DISCOS PUEDE GENERAR INEFICIENCIA, EN ESPECIAL EN SISTEMAS CON MULTIPROGRAMACION.
- v **HARDWARE PARA DISCOS.**
- v LOS DISCOS ESTAN ORGANIZADOS EN CILINDROS, PISTAS Y SECTORES.
- v EL N° TIPICO DE SECTORES POR PISTA VARIA ENTRE 8 Y 32 (O MAS).
- v TODOS LOS SECTORES TIENEN IGUAL N° DE BYTES.
- v LOS SECTORES CERCANOS A LA ORILLA DEL DISCO SERAN MAYORES FISICAMENTE QUE LOS CERCANOS AL ANILLO.
- v UN CONTROLADOR PUEDE REALIZAR BUSQUEDAS EN UNA O MAS UNIDADES AL MISMO TIEMPO:
 - υ SON LAS **BUSQUEDAS TRASLAPADAS.**
 - υ MIENTRAS EL CONTROLADOR Y EL SOFTWARE ESPERAN EL FIN DE UNA BUSQUEDA EN UNA UNIDAD:
 - Φ EL CONTROLADOR PUEDE INICIAR UNA BUSQUEDA EN OTRA.
- v MUCHOS CONTROLADORES PUEDEN:
 - υ LEER O ESCRIBIR EN UNA UNIDAD.
 - υ BUSCAR EN OTRA.
- v LOS CONTROLADORES NO PUEDEN LEER O ESCRIBIR EN DOS UNIDADES AL MISMO TIEMPO.
- v LA CAPACIDAD DE BUSQUEDAS TRASLAPADAS PUEDE REDUCIR CONSIDERABLEMENTE EL TIEMPO PROMEDIO DE ACCESO.

OPERACION DE ALMACENAMIENTO DE DISCO DE CABEZA MOVIL

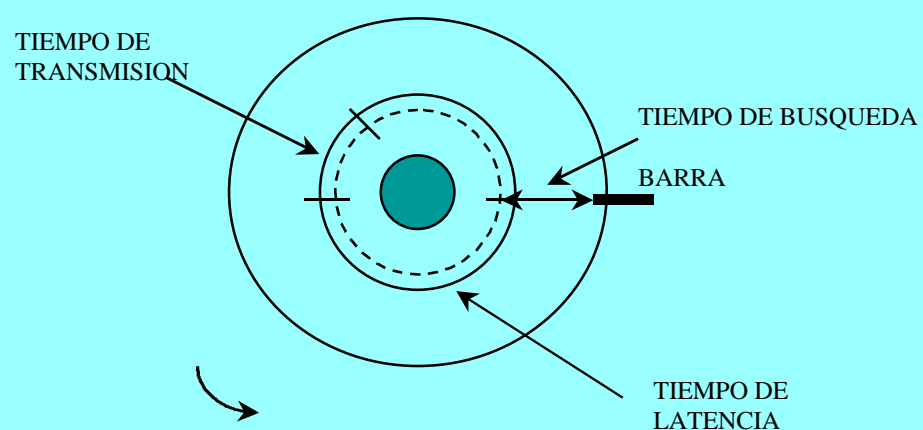
- v LOS DATOS SE GRABAN EN UNA SERIE DE DISCOS MAGNETICOS O *PLATOS*.
- v EL *EJE* COMUN DE LOS DISCOS GIRA A UNA VELOCIDAD DEL ORDEN DE LAS 3.600 REVOLUCIONES POR MINUTO.
- v SE LEE O ESCRIBE MEDIANTE UNA SERIE DE *CABEZAS DE LECTURA - ESCRITURA*:
 - u SE DISPONE DE UNA POR CADA SUPERFICIE DE DISCO.
 - u SOLO PUEDE ACCEDER A DATOS INMEDIATAMENTE ADYACENTES A ELLA:
 - Φ LA PARTE DE LA SUPERFICIE DEL DISCO DE DONDE SE LEERA (O SOBRE LA QUE SE GRABARA) DEBE *ROTAR* HASTA SITUARSE INMEDIATAMENTE DEBAJO (O ARRIBA) DE LA CABEZA DE LECTURA - ESCRITURA:
 - EL *TIEMPO DE ROTACION* DESDE LA POSICION ACTUAL HASTA LA ADYACENTE AL CABEZAL SE LLAMA *TIEMPO DE LATENCIA*.
- v TODAS LAS CABEZAS DE LECTURA - ESCRITURA ESTAN MONTADAS SOBRE UNA *BARRA O CONJUNTO DE BRAZO MOVIL*:
 - u PUEDE MOVERSE HACIA ADENTRO O HACIA AFUERA:
 - Φ OPERACION DE *BUSQUEDA*.
 - u PARA UNA POSICION DADA, LA SERIE DE *PISTAS* ACCESIBLES FORMAN UN *CILINDRO* VERTICAL.
- v A LOS TIEMPOS DE *BUSQUEDA* Y DE *LATENCIA* SE DEBE AGREGAR EL *TIEMPO DE TRANSMISION* PROPIAMENTE DICHA.
- v EL *TIEMPO TOTAL DE ACCESO* A UN REGISTRO PARTICULAR:
 - u INVOLUCRA MOVIMIENTOS MECANICOS.
 - u GENERALMENTE ES DEL ORDEN DE CENTESIMAS DE SEGUNDO, AUNQUE EL TIEMPO DE LATENCIA SEA DE ALGUNAS MILESIMAS DE SEGUNDO (7 A 12 APROXIMADAMENTE).

OPERACION DE ALMACENAMIENTO DE DISCO DE CABEZA MOVIL

ESQUEMA DE UN DISCO DE CABEZA MOVIL



COMPONENTES DEL ACCESO A UN DISCO



ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO

- v EN LA MAYORIA DE LOS DISCOS, EL TIEMPO DE BUSQUEDA SUPERA AL DE RETRASO ROTACIONAL Y AL DE TRANSFERENCIA:
 - u LA REDUCCION DEL TIEMPO PROMEDIO DE BUSQUEDA PUEDE MEJORAR EN GRAN MEDIDA EL RENDIMIENTO DEL SISTEMA.
- v SI EL MANEJADOR DEL DISCO UTILIZA EL ALGORITMO **PRIMERO EN LLEGAR PRIMERO EN SER ATENDIDO (FCFS)**:
 - u POCO SE PUEDE HACER PARA MEJORAR EL TIEMPO DE BUSQUEDA.
- v ES POSIBLE QUE MIENTRAS EL BRAZO REALIZA UNA BUSQUEDA PARA UNA SOLICITUD, OTROS PROCESOS GENEREN OTRAS SOLICITUDES.
- v MUCHOS MANEJADORES TIENEN UNA TABLA:
 - u EL INDICE ES EL N° DE CILINDRO.
 - u INCLUYE LAS SOLICITUDES PENDIENTES PARA CADA CILINDRO ENLAZADAS ENTRE SI EN UNA LISTA LIGADA.
 - u CUANDO CONCLUYE UNA BUSQUEDA, EL MANEJADOR DEL DISCO TIENE LA OPCION DE ELEGIR LA SIGUIENTE SOLICITUD A DAR PASO:
 - Φ SE ATIENDE PRIMERO LA SOLICITUD MAS CERCANA, PARA MINIMIZAR EL TIEMPO DE BUSQUEDA.
 - Φ ESTE ALGORITMO SE DENOMINA **PRIMERO LA BUSQUEDA MAS CORTA (SSF: SHORTEST SEEK FIRST)**.
 - Φ REDUCE A LA MITAD EL N° DE MOVIMIENTOS DEL BRAZO EN COMPARACION CON FCFS.
- v EJ. DE SSF:
 - u CONSIDERAMOS UN DISCO DE 40 CILINDROS.
 - u SE PRESENTA UNA SOLICITUD DE LECTURA DE UN BLOQUE EN EL CILINDRO 11.
 - u DURANTE LA BUSQUEDA, LLEGAN SOLICITUDES PARA LOS CILINDROS 1, 36, 16, 34, 9 Y 12, EN ESE ORDEN.
 - u LA SECUENCIA DE BUSQUEDA SSF SERA: 12, 9, 16, 1, 34, 36.
 - u N° DE MOVIMIENTOS DEL BRAZO PARA UN TOTAL DE 111 CILINDROS SEGUN FCFS.
 - u N° DE MOVIMIENTOS DEL BRAZO PARA UN TOTAL DE 61 CILINDROS SEGUN SSF.

ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO

- v EL ALGORITMO **SSF** TIENE EL SIGUIENTE PROBLEMA:
 - υ EL INGRESO DE NUEVAS SOLICITUDES PUEDE DEMORAR LA ATENCION DE LAS MAS ANTIGUAS.
 - υ CON UN DISCO MUY CARGADO, EL BRAZO TENDRA A PERMANECER A LA MITAD DEL DISCO LA MAYORIA DEL TIEMPO:
 - Φ LAS SOLICITUDES LEJANAS A LA MITAD DEL DISCO TENDRAN UN MAL SERVICIO.
 - υ ENTRAN EN CONFLICTO LOS OBJETIVOS DE:
 - Φ TIEMPO MINIMO DE RESPUESTA.
 - Φ JUSTICIA EN LA ATENCION.
- v LA SOLUCION A ESTE PROBLEMA LA BRINDA EL **ALGORITMO DEL ELEVADOR** (POR SU ANALOGIA CON EL ASCENSOR O ELEVADOR):
 - υ SE MANTIENE EL MOVIMIENTO DEL BRAZO EN LA MISMA DIRECCION, HASTA QUE NO TIENE MAS SOLICITUDES PENDIENTES EN ESA DIRECCION; ENTONCES CAMBIA DE DIRECCION.
 - υ EL SOFTWARE DEBE CONSERVAR EL BIT DE DIRECCION ACTUAL.
- v EJ. DEL ALGORITMO DEL ELEVADOR PARA EL CASO ANTERIOR, CON EL VALOR INICIAL *ARRIBA* DEL BIT DE DIRECCION:
 - υ EL ORDEN DE SERVICIO A LOS CILINDROS ES: 12, 16, 34, 36, 9 Y 1.
 - υ EL N° DE MOVIMIENTOS DEL BRAZO CORRESPONDE A 60 CILINDROS.
- v EL ALGORITMO DEL ELEVADOR:
 - υ OCASIONALMENTE ES MEJOR QUE EL ALGORITMO SSF.
 - υ GENERALMENTE ES PEOR QUE SSF.
 - υ DADA CUALQUIER COLECCION DE SOLICITUDES, LA CUOTA MAXIMA DEL TOTAL DE MOVIMIENTOS ESTA FIJA:
 - Φ ES EL DOBLE DEL N° DE CILINDROS.
- v UNA VARIANTE CONSISTE EN RASTREAR SIEMPRE EN LA MISMA DIRECCION:
 - υ LUEGO DE SERVIR AL CILINDRO CON EL N° MAYOR:
 - Φ EL BRAZO PASA AL CILINDRO DE N° MENOR CON UNA SOLICITUD PENDIENTE.
 - Φ CONTINUA SU MOVIMIENTO HACIA ARRIBA.

ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO

- v ALGUNOS CONTROLADORES DE DISCO PERMITEN QUE EL SOFTWARE INSPECCIONE EL N° DEL SECTOR ACTIVO DEBAJO DEL CABEZAL:
 - υ SI DOS O MAS SOLICITUDES PARA EL MISMO CILINDRO ESTAN PENDIENTES:
 - Φ EL MANEJADOR PUEDE ENVIAR UNA SOLICITUD PARA EL SECTOR QUE PASARA DEBAJO DEL CABEZAL.
 - Φ SE PUEDEN HACER SOLICITUDES CONSECUTIVAS DE DISTINTAS PISTAS DE UN MISMO CILINDRO, SIN GENERAR UN MOVIMIENTO DEL BRAZO.
- v CUANDO EXISTEN VARIAS UNIDADES, SE DEBE TENER UNA TABLA DE SOLICITUDES PENDIENTES PARA CADA UNIDAD.
- v SI UNA UNIDAD ESTA INACTIVA, DEBERA BUSCARSE EL CILINDRO SIGUIENTE NECESARIO:
 - υ SI EL CONTROLADOR PERMITE BUSQUEDAS TRASLAPADAS.
- v CUANDO TERMINA LA TRANSFERENCIA ACTUAL SE VERIFICA SI LAS UNIDADES ESTAN EN LA POSICION DEL CILINDRO CORRECTO:
 - υ SI UNA O MAS UNIDADES LO ESTAN, SE PUEDE INICIAR LA SIGUIENTE TRANSFERENCIA EN UNA UNIDAD YA POSICIONADA.
 - υ SI NINGUNO DE LOS BRAZOS ESTA POSICIONADO, EL MANEJADOR:
 - Φ DEBE REALIZAR UNA NUEVA BUSQUEDA EN LA UNIDAD QUE TERMINO LA TRANSFERENCIA.
 - Φ DEBE ESPERAR HASTA LA SIGUIENTE INTERRUPCION PARA VER CUAL BRAZO SE POSICIONA PRIMERO.
- v GENERALMENTE, LAS MEJORAS TECNOLOGICAS DE LOS DISCOS:
 - υ ACORTAN LOS TIEMPOS DE BUSQUEDA (SEEK).
 - υ NO ACORTAN LOS TIEMPOS DE DEMORA ROTACIONAL (SEARCH).
 - υ EN ALGUNOS DISCOS, EL TIEMPO PROMEDIO DE BUSQUEDA YA ES MENOR QUE EL RETRASO ROTACIONAL.
 - υ EL FACTOR DOMINANTE SERA EL RETRASO ROTACIONAL:
 - Φ LOS ALGORITMOS QUE OPTIMIZAN LOS TIEMPOS DE BUSQUEDA (COMO EL A. DEL ELEVADOR) PERDERAN IMPORTANCIA FRENTE A LOS A. QUE OPTIMICEN EL RETRASO ROTACIONAL.

ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO

- v UNA TECNOLOGIA IMPORTANTE ES LA QUE PERMITE EL TRABAJO CONJUNTO DE VARIOS DISCOS.
- v UNA CONFIGURACION INTERESANTE ES LA DE 38 UNIDADES EJECUTANDOSE EN PARALELO.
- v CUANDO SE REALIZA UNA OPERACION DE LECTURA:
 - u INGRESAN A LA CPU 38 BIT A LA VEZ, UNO POR CADA UNIDAD.
 - u LOS 38 BITS CONFORMAN UNA PALABRA DE 32 BITS JUNTO CON 6 BITS PARA VERIFICACION.
 - u LOS BITS 1, 2, 4, 8, 16 Y 32 SE UTILIZAN COMO BITS DE PARIDAD.
 - u LA PALABRA DE 38 BITS SE PUEDE CODIFICAR MEDIANTE EL **CODIGO HAMMING**:
 - Φ ES UN CODIGO CORRECTOR DE ERRORES.
 - u SI UNA UNIDAD SALE DE SERVICIO:
 - Φ SE PIERDE UN BIT DE CADA PALABRA.
 - Φ EL SISTEMA PUEDE CONTINUAR TRABAJANDO:
 - SE DEBE A QUE LOS CODIGOS HAMMING SE PUEDEN RECUPERAR DE UN BIT PERDIDO.
- v ESTE DISEÑO SE CONOCE COMO **RAID**:
 - u SIGLAS EN INGLES DE **ARREGLO REDUNDANTE DE DISCOS NO COSTOSOS**.

PORQUE ES NECESARIA LA PLANIFICACION DE DISCOS

- v EN LOS SISTEMAS DE MULTIPROGRAMACION MUCHOS PROCESOS PUEDEN ESTAR GENERANDO PETICIONES DE E / S SOBRE DISCOS:
 - u LA GENERACION DE PETICIONES PUEDE SER MUCHO MAS RAPIDA QUE LA ATENCION DE LAS MISMAS:
 - Φ SE CONSTRUYEN LINEAS DE ESPERA O COLAS PARA CADA DISPOSITIVO.
 - Φ PARA REDUCIR EL TIEMPO DE BUSQUEDA DE REGISTROS SE ORDENA LA COLA DE PETICIONES:
 - ESTO SE DENOMINA *PLANIFICACION DE DISCO*.
- v LA PLANIFICACION DE DISCO IMPLICA:
 - u UN EXAMEN CUIDADOSO DE LAS PETICIONES PENDIENTES PARA DETERMINAR LA FORMA MAS EFICIENTE DE SERVIRLAS.
 - u UN ANALISIS DE LAS RELACIONES POSICIONALES ENTRE LAS PETICIONES EN ESPERA.
 - u UN REORDENAMIENTO DE LA COLA DE PETICIONES PARA SERVIRLAS MINIMIZANDO LOS MOVIMIENTOS MECANICOS.
- v LOS TIPOS MAS COMUNES DE PLANIFICACION SON:
 - u OPTIMIZACION DE LA BUSQUEDA.
 - u OPTIMIZACION ROTACIONAL (LATENCIA).
- v GENERALMENTE LOS TIEMPOS DE BUSQUEDA SUPERAN A LOS DE LATENCIA, AUNQUE LA DIFERENCIA DISMINUYE:
 - u MUCHOS ALGORITMOS DE PLANIFICACION SE CONCENTRAN EN LA REDUCCION DE LOS TIEMPOS DE BUSQUEDA PARA UN CONJUNTO DE PETICIONES.
 - u GENERALMENTE LA REDUCCION DE LA LATENCIA RECIEN TIENE EFECTOS BAJO CARGAS DE TRABAJO MUY PESADAS.
- v BAJO CONDICIONES DE CARGA LIGERA (PROMEDIO BAJO DE LONGITUD DE LA COLA):
 - u ES ACEPTABLE EL DESEMPEÑO DEL METODO FCFS (PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO).
- v BAJO CONDICIONES DE CARGA MEDIA O PESADA:
 - u ES RECOMENDABLE UN ALGORITMO DE PLANIFICACION DE LAS

COLAS DE REQUERIMIENTOS.
ENTRADA / SALIDA

CARACTERISTICAS DESEABLES DE LAS POLITICAS DE PLANIFICACION DE DISCOS

- v LOS PRINCIPALES CRITERIOS DE CATEGORIZACION DE LAS POLITICAS DE PLANIFICACION SON:
 - u *CAPACIDAD DE EJECUCION.*
 - u *MEDIA DEL TIEMPO DE RESPUESTA.*
 - u *VARIANZA DE LOS TIEMPOS DE RESPUESTA (PREDECIBILIDAD).*
- v UNA POLITICA DE PLANIFICACION DEBE INTENTAR MAXIMIZAR LA *CAPACIDAD DE EJECUCION*:
 - u MAXIMIZAR EL N° DE PETICIONES SERVIDAS POR UNIDAD DE TIEMPO.
 - u MINIMIZAR LA MEDIA DEL TIEMPO DE RESPUESTA.
 - u MEJORAR EL RENDIMIENTO GLOBAL, QUIZAS A COSTA DE LAS PETICIONES INDIVIDUALES.
- v *LA PLANIFICACION SUELE MEJORAR LA IMAGEN TOTAL AL TIEMPO QUE REDUCE LOS NIVELES DE SERVICIO DE CIERTAS PETICIONES*:
 - u SE MIDE UTILIZANDO LA VARIANZA DE LOS TIEMPOS DE RESPUESTA.
 - u LA VARIANZA ES UN TERMINO ESTADISTICO QUE INDICA HASTA QUE PUNTO TIENDEN A DESVIARSE DEL PROMEDIO DE TODOS LOS ELEMENTOS LOS ELEMENTOS INDIVIDUALES.
 - u A MENOR VARIANZA MAYOR PREDECIBILIDAD.
 - u SE DESEA UNA POLITICA DE PLANIFICACION QUE MINIMICE LA VARIANZA, ES DECIR QUE MAXIMICE LA PREDECIBILIDAD.
 - u NO DEBE HABER PETICIONES QUE PUEDAN EXPERIMENTAR NIVELES DE SERVICIO ERRATICOS.

OPTIMIZACION DE LA BUSQUEDA EN DISCOS

- v LAS ESTRATEGIAS MAS COMUNES DE OPTIMIZACION DE LA BUSQUEDA SON LAS SIGUIENTES:
 - u FCFS.
 - u SSTF.
 - u SCAN.
 - u SCAN DE N-PASOS.
 - u C-SCAN.
 - u ESQUEMA ESCHENBACH.
- v **PLANIFICACION FCFS (PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO).**
- v UNA PETICION NO PUEDE SER DESPLAZADA POR LA LLEGADA DE UNA PETICION CON PRIORIDAD MAS ALTA.
- v NO HAY REORDENAMIENTO DE LA COLA DE PETICIONES PENDIENTES.
- v SE IGNORAN LAS RELACIONES POSICIONALES ENTRE LAS PETICIONES PENDIENTES.
- v OFRECE UNA VARIANZA PEQUEÑA AUNQUE PERJUDICA A LAS PETICIONES SITUADAS AL FINAL DE LA COLA.
- v **PLANIFICACION SSTF (MENOR TIEMPO DE BUSQUEDA PRIMERO).**
- v EL BRAZO DEL DISCO SE SITUA EN LA SIGUIENTE PETICION QUE MINIMICE EL MOVIMIENTO DEL BRAZO.
- v NO RESPETA EL ORDEN DE LLEGADA DE LAS PETICIONES A LA COLA.
- v TIENDE A FAVORECER A LAS PISTAS DEL CENTRO DEL DISCO.
- v LA MEDIA DE TIEMPOS DE RESPUESTA TIENDE A SER MAS BAJA QUE CON FCFS, PARA CARGAS MODERADAS.
- v LAS VARIANZAS TIENDEN A SER MAYORES QUE CON FCFS POR EL EFECTO DE LAS PISTAS INTERIORES Y EXTERIORES.
- v **PLANIFICACION SCAN.**
- v EL BRAZO DEL DISCO SE DESPLAZA SIRVIENDO A TODAS LAS PETICIONES QUE ENCUENTRA A SU PASO.
- v CAMBIA DE DIRECCION CUANDO YA NO HAY PETICIONES PENDIENTES EN LA DIRECCION ACTUAL.
- v HA SIDO LA BASE DE LA MAYORIA DE LAS ESTRATEGIAS DE PLANIFICACION IMPLEMENTADAS.
- v ELIMINA LAS DISCRIMINACIONES DE SSTF Y TIENE MENOR VARIANZA.
- v LAS PISTAS EXTERIORES SON MENOS VISITADAS QUE LAS INTERMEDIAS, PERO NO ES TAN GRAVE COMO CON SSTF.

OPTIMIZACION DE LA BUSQUEDA EN DISCOS

- v **PLANIFICACION SCAN DE N-PASOS.**
- v LA ESTRATEGIA DE MOVIMIENTO DEL BRAZO ES COMO EN SCAN.
- v SOLO DA SERVICIO A LAS PETICIONES QUE SE ENCUENTRAN EN ESPERA CUANDO COMIENZA UN RECORRIDO PARTICULAR.
- v LAS PETICIONES QUE LLEGAN DURANTE UN RECORRIDO SON AGRUPADAS Y ORDENADAS:
 - u SERAN ATENDIDAS DURANTE EL RECORRIDO DE REGRESO.
- v POSEE MENOR VARIANZA DE LOS TIEMPOS DE RESPUESTA SI SE COMPARA CON LAS PLANIFICACIONES SSTF Y SCAN CONVENCIONALES.
- v **PLANIFICACION C-SCAN (BUSQUEDA CIRCULAR).**
- v EL BRAZO SE MUEVE DEL CILINDRO EXTERIOR AL INTERIOR, SIRVIENDO A LAS PETICIONES SOBRE UNA BASE DE BUSQUEDA MAS CORTA.
- v FINALIZADO EL RECORRIDO HACIA EL INTERIOR, SALTA A LA PETICION MAS CERCANA AL CILINDRO EXTERIOR Y REANUDA SU DESPLAZAMIENTO HACIA EL INTERIOR.
- v NO DISCRIMINA A LOS CILINDROS EXTERIOR E INTERIOR.
- v LA VARIANZA DE LOS TIEMPOS DE RESPUESTA ES MUY PEQUEÑA.
- v **ESQUEMA ESCHENBACH.**
- v EL BRAZO DEL DISCO SE MUEVE COMO EN C-SCAN, PERO:
 - u LAS PETICIONES SE REORDENAN PARA SER SERVIDAS DENTRO DE UN CILINDRO PARA TOMAR VENTAJA DE LA POSICION ROTACIONAL.
 - u SI DOS PETICIONES TRASLADAN POSICIONES DE SECTORES DENTRO DE UN CILINDRO:
 - Φ SOLO SE SIRVE UNA EN EL MOVIMIENTO ACTUAL DEL BRAZO DEL DISCO.
- v ESTA ESTRATEGIA TIENE EN CUENTA EL RETRASO ROTACIONAL.
- v **CONCLUSIONES.**
- v MEDIANTE TRABAJOS DE SIMULACION Y DE LABORATORIO SE DEMOSTRO LO SIGUIENTE:
 - u LA ESTRATEGIA SCAN ES LA MEJOR CON CARGA BAJA.
 - u LA ESTRATEGIA C-SCAN ES LA MEJOR CON CARGAS MEDIAS Y PESADAS.
 - u LA ESTRATEGIA C-SCAN CON OPTIMIZACION ROTACIONAL ES LA MEJOR PARA CARGAS MUY PESADAS (MEJOR QUE LA ESTRATEGIA

OPTIMIZACION ROTACIONAL EN DISCOS

CONSIDERACIONES DE LOS DISCOS SOBRE LOS SISTEMAS

- v **OPTIMIZACION ROTACIONAL EN DISCOS.**
- v EN CONDICIONES DE CARGA PESADA, LAS PROBABILIDADES DE QUE OCURRAN REFERENCIAS AL MISMO CILINDRO AUMENTAN:
 - u RESULTA UTIL CONSIDERAR LA OPTIMIZACION ROTACIONAL ADEMAS DE LA OPTIMIZACION DE BUSQUEDA.
- v LA OPTIMIZACION ROTACIONAL ES DE USO COMUN EN DISPOSITIVOS DE CABEZAS FIJAS.
- v LA ESTRATEGIA UTILIZADA ES LA **SLTF (TIEMPO DE LATENCIA MAS CORTO PRIMERO)**:
 - u SITUADO EL BRAZO DEL DISCO EN UN CILINDRO:
 - φ EXAMINA TODAS LAS PETICIONES SOBRE EL CILINDRO.
 - φ SIRVE PRIMERO A LA QUE TIENE EL RETRASO ROTACIONAL MAS CORTO.
- v **CONSIDERACIONES DE LOS DISCOS SOBRE LOS SISTEMAS.**
- v LOS PRINCIPALES INTERROGANTES SON:
 - u CUANDO ES UTIL LA PLANIFICACION DE DISCO.
 - u CUANDO PUEDE DEGRADAR EL RENDIMIENTO.
- v EL ALMACENAMIENTO EN DISCO COMO UN RECURSO LIMITADOR.
- v LA PLANIFICACION DE DISCO PUEDE MEJORAR EL RENDIMIENTO Y ELIMINAR EL *EMBOTELLAMIENTO*:
 - u EL EMBOTELLAMIENTO SE PRODUCE CUANDO SE CONCENTRAN GRANDES CARGAS DE PETICIONES SOBRE RELATIVAMENTE POCOS DISCOS O POCOS CILINDROS DE UN DISCO.
- v NIVEL DE MULTIPROGRAMACION.
- v GENERALMENTE LA PLANIFICACION ES EFECTIVA EN SISTEMAS DE TIEMPO COMPARTIDO CON UN NIVEL ALTO DE MULTIPROGRAMACION.
- v SUBSISTEMAS DE DISCOS MULTIPLES.
- v FRECUENTEMENTE LA CPU ESTA CONECTADA MEDIANTE CANALES (O BUS) A DISPOSITIVOS CONTROLADORES, LOS QUE ESTAN CONECTADOS A LAS UNIDADES DE DISCOS.
- v EL EMBOTELLAMIENTO PUEDE PRODUCIRSE EN ALGUN DISCO, ALGUN CONTROLADOR O EN ALGUN CANAL.
- v EXISTE SOFTWARE ESPECIFICO PARA:
 - u MEDIR LA ACTIVIDAD.
 - u DETECTAR DONDE SE PRODUCE EL EMBOTELLAMIENTO.

CONSIDERACIONES DE LOS DISCOS SOBRE LOS SISTEMAS

- v PARA ELIMINAR CIERTOS EMBOTELLAMIENTOS PUEDE SER NECESARIA UNA *RECONFIGURACION DEL HARDWARE*:
 - u AGREGAR CANALES, CONTROLADORES, DISPOSITIVOS.
 - u CAMBIAR DISPOSITIVOS DE UN CONTROLADOR A OTRO.
 - u CAMBIAR CONTROLADORES DE UN CANAL A OTRO.
- v PARA AYUDAR A REDUCIR LA CONGESTION DEL CANAL, MUCHOS SISTEMAS HAN INCORPORADO LA TECNICA DE *EXAMEN (SENSADO) DE POSICION ROTACIONAL (RPS)*:
 - u REDUCE EL TIEMPO DURANTE EL CUAL UN CANAL SE ENCUENTRA OCUPADO EN LA BUSQUEDA DE UN REGISTRO.
 - u RPS PERMITE AL CANAL QUEDAR LIBRE JUSTO HASTA ANTES DE QUE EL REGISTRO SE ENCUENTRE DEBAJO DE LA CABEZA DE LECTURA-GRABACION APROPIADA.
 - u RPS PERMITE VARIAS PETICIONES ACTIVAS AL MISMO TIEMPO EN UN SOLO CANAL, INCREMENTANDO LA PERFORMANCE.
- v DISTRIBUCION DE PETICIONES NO UNIFORMES.
- v SON MUY COMUNES EN CIERTAS SITUACIONES REALES.
- v SON FRECUENTES EN PROCESOS SECUENCIALES DE ARCHIVOS SECUENCIALES, PARA LOS QUE SE AFECTARON CILINDROS ADYACENTES INMEDIATOS.
- v GENERALMENTE EN ESTOS CASOS LAS BUSQUEDAS SON CORTAS Y LA PLANIFICACION DE DISCO SERA DE POCA UTILIDAD.
- v TECNICAS DE ORGANIZACION DE ARCHIVOS.
- v LOS METODOS DE ORGANIZACION Y ACCESO DE ARCHIVOS, ASI COMO LOS DBMS (MANEJADORES DE BASES DE DATOS):
 - u SON MUY CONVENIENTES DESDE EL PUNTO DE VISTA DE LAS APLICACIONES Y DEL USUARIO.
 - u PUEDEN GENERAR COMPLICACIONES EN LA IMPLEMENTACION Y EL RENDIMIENTO:
 - φ EL RECORRIDO DE ESTRUCTURAS DE INDICES, BLOQUES DE CONTROL, APUNTADORES, ETC., PUEDE SIGNIFICAR UN GRAN

Nº DE OPERACIONES DE E / S.

MANEJO DE ERRORES EN DISCOS

- v ALGUNOS DE LOS ERRORES MAS COMUNES EN DISCOS SON:
 - u ERROR DE PROGRAMACION:
 - Φ EJ.: SOLICITAR UN SECTOR NO EXISTENTE.
 - u ERROR TEMPORAL EN LA SUMA DE VERIFICACION:
 - Φ EJ.: PROVOCADO POR POLVO EN LA CABEZA.
 - u ERROR PERMANENTE EN LA SUMA DE VERIFICACION:
 - Φ EJ.: UN BLOQUE DEL DISCO DAÑADO FISICAMENTE.
 - u ERROR DE BUSQUEDA:
 - Φ EJ.: EL BRAZO SE ENVIA AL CILINDRO 6 PERO VA AL 7.
 - u ERROR DEL CONTROLADOR:
 - Φ EJ.: EL CONTROLADOR NO ACEPTA LOS COMANDOS.
- v EL MANEJADOR DEL DISCO DEBE CONTROLAR LOS ERRORES DE LA MEJOR MANERA POSIBLE.
- v LA MAYORIA DE LOS CONTROLADORES:
 - u VERIFICAN LOS PARAMETROS QUE SE LES PROPORCIONAN.
 - u INFORMAN SI NO SON VALIDOS.
- v RESPECTO DE LOS ERRORES TEMPORALES EN LA SUMA DE VERIFICACION:
 - u GENERALMENTE SE ELIMINAN AL REPETIR LA OPERACION.
 - u SI PERSISTEN, EL BLOQUE DEBE SER MARCADO COMO UN **BLOQUE DEFECTUOSO**, PARA QUE EL SOFTWARE LO EVITE.
- v OTRA POSIBILIDAD ES QUE CONTROLADORES “INTELIGENTES” RESERVEN CIERTA CANTIDAD DE PISTAS:
 - u SERAN ASIGNADAS EN REEMPLAZO DE PISTAS DEFECTUOSAS.
 - u UNA TABLA ASOCIA LAS PISTAS DEFECTUOSAS CON LAS PISTAS DE REPUESTO:
 - Φ ESTA ALOJADA EN LA MEMORIA INTERNA DEL CONTROLADOR Y EN EL DISCO.
 - Φ LA SUSTITUCION ES TRANSPARENTE PARA EL MANEJADOR.
 - Φ PUEDE AFECTARSE EL DESEMPEÑO DE LOS ALGORITMOS DE BUSQUEDA, COMO EL DEL ELEVADOR:
 - EL CONTROLADOR UTILIZA PISTAS FISICAMENTE DISTINTAS DE LAS SOLICITADAS.

OCULTAMIENTO DE UNA PISTA A LA VEZ EN DISCOS

- v GENERALMENTE EL TIEMPO DE BUSQUEDA SUPERA AL DE ROTACION Y TRANSFERENCIA (AUNQUE ESTO SE ESTA EQUILIBRANDO).
- v UNA VEZ RESUELTA LA BUSQUEDA DEL CILINDRO CORRESPONDIENTE, NO ES MUY IMPORTANTE SI SE LEE UN SECTOR O TODA LA PISTA:
 - u ESPECIALMENTE EN DISPOSITIVOS CON **SENSIBILIDAD ROTACIONAL (RPS)**:
 - ⊕ EL MANEJADOR PUEDE VER QUE SECTOR SE ENCUENTRA DEBAJO DE LA CABEZA Y PUEDE ENVIAR UNA SOLICITUD DEL SIGUIENTE SECTOR:
 - PERMITE LEER UNA PISTA EN UN TIEMPO DE ROTACION.
 - DE LO CONTRARIO SE TARDARIA, EN PROMEDIO, UN TIEMPO DE ROTACION MAS UN TIEMPO DE SECTOR, PARA LEER UN SOLO SECTOR.
 - u ALGUNOS MANEJADORES APROVECHAN ESTO MEDIANTE UN **CACHE SECRETO DE UNA PISTA A LA VEZ**:
 - ⊕ ES DESCONOCIDO POR EL SOFTWARE INDEPENDIENTE DEL DISPOSITIVO.
 - ⊕ SI SE NECESITA UN SECTOR DEL CACHE, NO ES NECESARIA UNA TRANSFERENCIA DEL DISCO.
 - ⊕ LAS PRINCIPALES DESVENTAJAS DE ESTE OCULTAMIENTO DE UNA PISTA A LA VEZ SON:
 - COMPLEJIDAD DEL SOFTWARE.
 - REQUERIMIENTOS DE ESPACIO PARA BUFFERS.
 - LAS TRANSFERENCIAS DEL CACHE AL PROGRAMA QUE HACE LA LLAMADA:
 - LAS DEBE REALIZAR LA CPU MEDIANTE UN CICLO PROGRAMADO.
 - NO LAS PUEDE HACER EL HARDWARE DMA.
 - ⊕ ALGUNOS CONTROLADORES REALIZAN EL OCULTAMIENTO DE UNA PISTA A LA VEZ EN SU PROPIA MEMORIA INTERNA:
 - RESULTA TRANSPARENTE AL MANEJADOR.
 - LAS TRANSFERENCIAS ENTRE EL CONTROLADOR Y LA MEMORIA PUEDEN UTILIZAR DMA.

RELOJES

- v **DISCOS EN RAM.**
- v UTILIZAN UNA PARTE DE LA MEMORIA PRINCIPAL ASIGNADA CON ANTERIORIDAD PARA ALMACENAR LOS BLOQUES.
- v TIENEN LA VENTAJA DEL ACCESO INSTANTANEO:
 - v NO HAY DEMORA ROTACIONAL O DEBIDA A LAS BUSQUEDAS.
 - v SON ADECUADOS PARA EL ALMACENAMIENTO DE PROGRAMAS O DATOS CON ACCESOS MUY FRECUENTES.
- v LOS BLOQUES DE ALMACENAMIENTO TIENEN EL MISMO TAMAÑO QUE EN LOS DISCOS REALES.
- v CUANDO EL MANEJADOR DEBE LEER DE O ESCRIBIR EN UN BLOQUE DE UN DISCO EN RAM:
 - v CALCULA EL LUGAR DE LA MEMORIA DONDE SE ENCUENTRA EL BLOQUE SOLICITADO Y LEE O ESCRIBE EN EL MISMO.
- v **RELOJES.**
- v LOS **RELOJES O CRONOMETROS** SON ESENCIALES PARA LA OPERACION DE SISTEMAS DE TIEMPO COMPARTIDO.
- v REGISTRAN LA HORA DEL DIA.
- v EVITAN QUE UN PROCESO MONOPOLICE LA CPU.
- v EL SOFTWARE PARA RELOJ TOMA GENERALMENTE LA FORMA DE UN MANEJADOR DE DISPOSITIVO, AUNQUE NO ES UN DISPOSITIVO DE BLOQUE NI DE CARACTER.
- v LOS RELOJES MAS SENCILLO TRABAJAN CON LA LINEA DE CORRIENTE ELECTRICA DE 110 O 220 VOLTIOS:
 - v PROVOCAN UNA INTERRUPCION POR CADA CICLO DE VOLTAJE, A 50 O 60 HZ.
- v OTRO TIPO DE RELOJES CONSTA DE TRES COMPONENTES:
 - v UN OSCILADOR DE CRISTAL, UN CONTADOR Y UN REGISTRO.
 - v UNA PIEZA DE CRISTAL DE CUARZO SE MONTA EN UNA ESTRUCTURA BAJO TENSION:
 - ❖ GENERA UNA SEÑAL PERIODICA DE MUY ALTA PRECISION, GENERALMENTE ENTRE 5 Y 100 MHZ.
 - ❖ LA SEÑAL SE ALIMENTA EN EL CONTADOR PARA QUE CUENTE EN FORMA DESCENDENTE HASTA CERO.
 - ❖ CUANDO EL CONTADOR LLEGA A CERO, PROVOCA UNA INTERRUPCION DE LA CPU.

RELOJES

- v LOS **RELOJES PROGRAMABLES** TIENEN VARIOS MODOS DE OPERACION:
 - u **MODO DE UNA INSTANCIA:**
 - Φ CUANDO EL RELOJ SE INICIALIZA, COPIA EL VALOR DEL REGISTRO EN EL CONTADOR.
 - Φ DECREMENTA EL CONTADOR EN CADA PULSO DEL CRISTAL.
 - Φ CUANDO EL CONTADOR LLEGA A CERO PROVOCA UNA INTERRUPCION Y SE DETIENE HASTA SER NUEVAMENTE INICIALIZADO POR EL SOFTWARE.
 - u **MODO DE ONDA CUADRADA:**
 - Φ LUEGO DE LLEGAR A CERO Y PROVOCAR LA INTERRUPCION, EL REGISTRO SE COPIA DE MANERA AUTOMATICA EN EL CONTADOR.
 - Φ TODO EL PROGRAMA SE REPITE EN FORMA INDEFINIDA.
 - Φ LAS INTERRUPCIONES PERIODICAS SE LLAMAN **MARCAS DEL RELOJ**.
- v LA VENTAJA DEL RELOJ PROGRAMABLE ES QUE SU FRECUENCIA DE INTERRUPCION PUEDE SER CONTROLADA POR EL SOFTWARE.
- v LAS PRINCIPALES FUNCIONES DEL SOFTWARE MANEJADOR DEL RELOJ SON:
 - u MANTENER LA HORA DEL DIA O **TIEMPO REAL**.
 - u EVITAR QUE LOS PROCESOS SE EJECUTEN DURANTE MAS TIEMPO DEL PERMITIDO.
 - u MANTENER UN REGISTRO DEL USO DE LA CPU.
 - u CONTROLAR LLAMADAS AL SISTEMA TIPO **ALARM** POR PARTE DE LOS PROCESOS DEL USUARIO.
 - u PROPORCIONAR CRONOMETROS GUARDIANES DE PARTES DEL PROPIO SISTEMA.
 - u REALIZAR RESUMENES, MONITOREO Y RECOLECCION DE ESTADISTICAS.
- v EL SOFTWARE MANEJADOR DEL RELOJ PUEDE TENER QUE SIMULAR VARIOS RELOJES VIRTUALES CON UN UNICO RELOJ FISICO.

TERMINALES

- v LAS TERMINALES TIENEN GRAN N° DE FORMAS DISTINTAS:
 - u EL MANEJADOR DE LA TERMINAL DEBE OCULTAR ESTAS DIFERENCIAS:
 - Φ LA PARTE INDEPENDIENTE DEL DISPOSITIVO EN EL S. O. Y LOS PROGRAMAS DEL USUARIO NO SE TIENEN QUE REESCRIBIR PARA CADA TIPO DE TERMINAL.
- v DESDE EL PUNTO DE VISTA DEL S. O. SE LAS PUEDE CLASIFICAR EN:
 - u INTERFAZ RS-232:
 - Φ HARDCOPY (TERMINALES DE IMPRESION).
 - Φ TTY “DE VIDRIO” (TERMINALES DE VIDEO).
 - Φ INTELIGENTE (COMPUTADORAS CON CPU Y MEMORIA).
 - u INTERFAZ MAPEADA A MEMORIA:
 - Φ ORIENTADA A CARACTERES.
 - Φ ORIENTADA A BITS.
- v LAS TERMINALES RS-232 POSEEN UN TECLADO Y UN MONITOR QUE SE COMUNICAN MEDIANTE UNA INTERFAZ SERIAL, UN BIT A LA VEZ:
 - u LAS CONVERSIONES DE BITS A BYTES Y VICEVERSA LAS EFECTUAN LOS CHIPS UART (**TRANSMISORES-RECEPTORES ASINCRONOS UNIVERSALES**).
- v LAS TERMINALES MAPEADAS A MEMORIA:
 - u NO SE COMUNICAN MEDIANTE UNA LINEA SERIAL.
 - u POSEEN UNA INTERFAZ MEDIANTE UNA MEMORIA ESPECIAL LLAMADA **VIDEO RAM**:
 - Φ FORMA PARTE DEL ESPACIO DE DIRECCIONES DE LA COMPUTADORA.
 - Φ LA CPU SE DIRIGE A ELLA COMO AL RESTO DE LA MEMORIA.
 - Φ EN LA TARJETA DE VIDEO RAM HAY UN CHIP LLAMADO **CONTROLADOR DE VIDEO**:
 - EXTRAE BYTES DEL VIDEO RAM Y GENERA LA SEÑAL DE VIDEO UTILIZADA PARA MANEJAR LA PANTALLA.
 - EL MONITOR GENERA UN RAYO DE ELECTRONES QUE RECORRE LA PANTALLA PINTANDO LINEAS.
 - CADA LINEA ESTA CONSTITUIDA POR UN CIERTO N° DE PUNTOS O **PIXELES**.
 - LA SEÑAL DEL CONTROLADOR DE VIDEO MODULA EL RAYO DE ELECTRONES Y DETERMINA SI UN PIXEL DEBE ESTAR O NO ILUMINADO.

TERMINALES

- LOS MONITORES DE COLOR POSEEN TRES RAYOS (ROJO, VERDE Y AZUL) QUE SE MODULAN INDEPENDIENTEMENTE.
- v EN LAS PANTALLAS MAPEADAS A CARACTERES:
 - u CADA CARACTER EN LA PANTALLA EQUIVALE A DOS CARACTERES DE RAM:
 - Φ UNO ALOJA AL CODIGO (ASCII) DEL CARACTER POR EXHIBIR.
 - Φ OTRO ES EL BYTE DE ATRIBUTO, NECESARIO PARA DETERMINAR EL COLOR, EL VIDEO INVERSO, EL PARPADEO, ETC.
- v EN LAS TERMINALES MAPEADAS A BITS:
 - u SE UTILIZA EL MISMO PRINCIPIO.
 - u CADA BIT EN EL VIDEO RAM CONTROLA EN FORMA DIRECTA UN SOLO PIXEL DE LA PANTALLA.
 - u PERMITE UNA COMPLETA FLEXIBILIDAD EN LOS TIPOS Y TAMAÑOS DE CARACTERES, VARIAS VENTANAS Y GRAFICOS ARBITRARIOS.
- v CON LAS PANTALLAS MAPEADAS A MEMORIA, EL TECLADO SE DESACOPLA TOTALMENTE DE LA PANTALLA:
 - u EL TECLADO DISPONE DE SU PROPIO MANEJADOR.
 - u EL MANEJADOR DEL TECLADO PUEDE OPERAR EN MODO CARACTER O EN MODO LINEA.
- v LAS TERMINALES PUEDEN OPERAR CON UNA ESTRUCTURA CENTRAL DE BUFFERS O CON BUFFERS EXCLUSIVOS PARA CADA TERMINAL.
- v FRECUENTEMENTE LOS MANEJADORES DE TERMINALES SOPORTAN OPERACIONES TALES COMO:
 - u MOVER EL CURSOR HACIA ARRIBA, ABAJO, A LA IZQUIERDA O A LA DERECHA UNA POSICION.
 - u MOVER EL CURSOR A X,Y.
 - u INSERTAR UN CARACTER O UNA LINEA EN EL CURSOR.
 - u ELIMINAR UN CARACTER O UNA LINEA EN EL CURSOR.
 - u RECORRER LA PANTALLA HACIA ARRIBA O HACIA ABAJO N LINEAS.
 - u LIMPIAR LA PANTALLA DESDE EL CURSOR HACIA EL FINAL DE LA LINEA O HASTA EL FINAL DE LA PANTALLA.
 - u TRABAJAR EN MODO DE VIDEO INVERSO, SUBRAYADO, PARPADEO O NORMAL.
 - u CREAR, CONSTRUIR, MOVER O CONTROLAR LAS VENTANAS.

BLOQUEOS

- v INTRODUCCION Y EJEMPLOS DE BLOQUEO (O INTERBLOQUEO)
- v CONCEPTOS DE RECURSOS
- v BLOQUEOS Y CONDICIONES NECESARIAS PARA EL BLOQUEO
- v MODELACION DE BLOQUEOS
- v AREAS PRINCIPALES EN LA INVESTIGACION DE BLOQUEOS
- v EL ALGORITMO DEL AVESTRUZ O DE OSTRICH
- v DETECCION DE BLOQUEOS
- v RECUPERACION DE BLOQUEOS
- v EVASION DE BLOQUEOS
- v PREVENCIÓN DE BLOQUEOS
- v OTROS ASPECTOS
- v TENDENCIAS DEL TRATAMIENTO DEL BLOQUEO

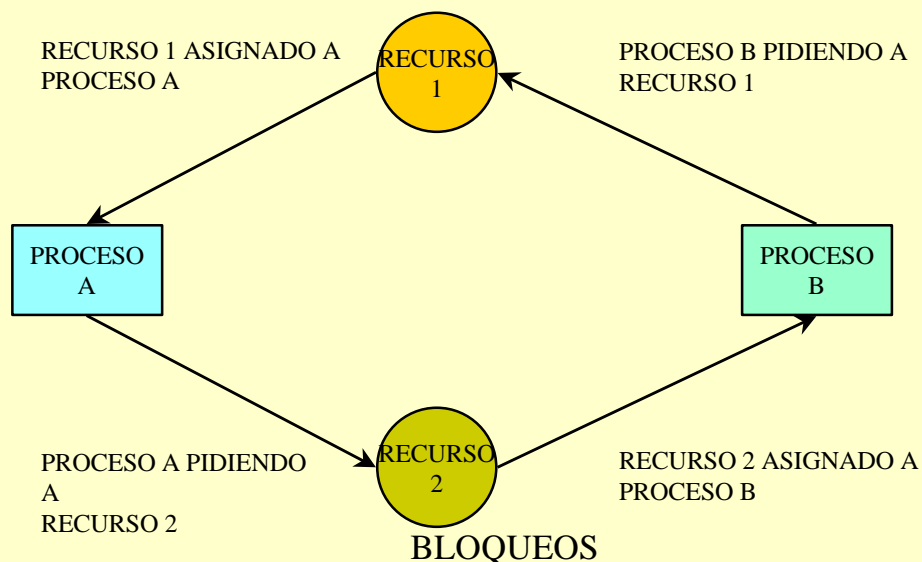
INTRODUCCION Y EJEMPLOS DE BLOQUEO (O INTERBLOQUEO)

- v UN PROCESO DENTRO DE UN SISTEMA DE MULTIPROGRAMACION ESTA EN UN ESTADO DE *INTERBLOQUEO* (O *INTERBLOQUEADO*) SI ESTA ESPERANDO POR UN EVENTO DETERMINADO QUE NO OCURRIRA.
- v CUANDO LOS RECURSOS SON COMPARTIDOS ENTRE USUARIOS:
 - u PUEDEN PRODUCIRSE *INTERBLOQUEOS* EN LOS CUALES LOS PROCESOS DE ALGUNOS USUARIOS NUNCA PODRAN LLEGAR A SU TERMINO.
 - u SE DEBE CONSIDERAR LA *PREVENCION*, *EVITACION*, *DETECCION* Y *RECUPERACION* DEL INTERBLOQUEO Y LA *POSTERGACION INDEFINIDA*:
 - Φ UN PROCESO, AUNQUE NO ESTE INTERBLOQUEADO, PUEDE ESTAR ESPERANDO POR UN EVENTO QUE PROBABLEMENTE NUNCA OCURRIRA.
 - u EN ALGUNOS CASOS:
 - Φ EL PRECIO DE LIBERAR INTERBLOQUEOS EN UN SISTEMA ES DEMASIADO ALTO.
 - Φ PERMITIR EL INTERBLOQUEO PODRIA RESULTAR CATASTROFICO.
- v LOS SISTEMAS DE COMPUTOS TIENEN MUCHOS RECURSOS QUE SOLO PUEDEN SER UTILIZADOS POR UN PROCESO A LA VEZ:
 - u EJ.: IMPRESORAS, UNIDADES DE CINTA, ESPACIO DE LA TABLA DE NODOS-I.
 - u LOS S. O. TIENEN LA CAPACIDAD DE OTORGAR TEMPORALMENTE A UN PROCESO EL *ACCESO EXCLUSIVO* A CIERTOS RECURSOS.
 - u FRECUENTEMENTE UN PROCESO NECESITA EL ACCESO EXCLUSIVO NO SOLO A UN RECURSO, SINO A VARIOS.
- v EJ. DE *BLOQUEO (DEADLOCK)*:
 - u DOS PROCESOS DESEAN IMPRIMIR GRANDES ARCHIVOS EN CINTA.
 - u EL PROCESO A SOLICITA LA IMPRESORA, QUE SE LE CONCEDE.
 - u EL PROCESO B SOLICITA LA UNIDAD DE CINTA, QUE SE LE CONCEDE.
 - u EL PROCESO A SOLICITA LA UNIDAD DE CINTA:
 - Φ SE DENIEGA LA SOLICITUD HASTA QUE B LA LIBERA.
 - u EL PROCESO B SOLICITA LA IMPRESORA:
 - Φ SE PRODUCE EL *BLOQUEO (DEADLOCK)*.

INTRODUCCION Y EJEMPLOS DE BLOQUEO (O INTERBLOQUEO)

- v **EJEMPLO DE INTERBLOQUEO DE TRAFICO:**
- v TIENE SIMILITUD CON EL CONGESTIONAMIENTO DEL TRANSITO EN LAS CIUDADES.
- v EL TRAFICO PUEDE DETENERSE COMPLETAMENTE.
- v ES NECESARIA UNA INTERVENCION EXTERNA PARA PONER ORDEN Y RESTABLECER LA NORMALIDAD.
- v **EJEMPLO DE INTERBLOQUEO DE UN RECURSO SIMPLE:**
- v TIENE SU ORIGEN EN LA CONTENCIÓN NORMAL DE LOS *RECURSOS DEDICADOS O REUTILIZABLES EN SERIE*:
 - v PUEDEN SER UTILIZADOS POR UN SOLO USUARIO A LA VEZ.
 - v CADA PROCESO ESTA ESPERANDO POR EL OTRO PARA LIBERAR UNO DE LOS RECURSOS.
 - v EL RECURSO RETENIDO NO SERA LIBERADO HASTA QUE EL OTRO PROCESO USUARIO LIBERE SU RECURSO.
 - v ESTE ULTIMO PROCESO USUARIO NO LIBERARA SU RECURSO RETENIDO HASTA QUE EL PRIMER PROCESO USUARIO LIBERE SU RECURSO RETENIDO.
 - v SE PRODUCE UNA *ESPERA CIRCULAR*.

UN INTERBLOQUEO SIMPLE



INTRODUCCION Y EJEMPLOS DE BLOQUEO (O INTERBLOQUEO)

- v **EJEMPLO DE INTERBLOQUEO EN SISTEMAS DE SPOOL:**
- v UN SISTEMA DE SPOOL ES UTILIZADO PARA INCREMENTAR LA CAPACIDAD DE EJECUCION DEL SISTEMA, AL DISASOCIAR UN PROGRAMA DE LA LENTA VELOCIDAD DE LOS DISPOSITIVOS (EJ.: IMPRESORAS):
 - v SI UN PROGRAMA ENVIA LINEAS A UNA IMPRESORA, EN REALIDAD SON ENVIADAS A UN DISPOSITIVO MAS RAPIDO (DISCO).
 - v SE ALMACENAN TEMPORALMENTE HASTA SER IMPRESAS.
- v VARIOS TRABAJOS EN EJECUCION QUE GENERAN LINEAS DE SPOOL PUEDEN INTERBLOQUEARSE SI EL ESPACIO DISPONIBLE SE LLENA ANTES DE COMPLETARSE ALGUNO DE ESTOS TRABAJOS:
 - v SE REDUCE LA PROBABILIDAD DE INTERBLOQUEOS DEL SPOOL:
 - ⊕ PROPORCIONANDO UN ESPACIO EN DISCO CONSIDERABLEMENTE MAYOR QUE EL NECESARIO, PREFERENTEMENTE CON ASIGNACION DINAMICA.
 - ⊕ LIMITANDO LOS *SPOOLERS* DE ENTRADA PARA QUE NO LEAN MAS TRABAJOS CUANDO LOS ARCHIVOS DE *SPOOL* LLEGAN A CIERTO NIVEL DE SATURACION.
- v **UN PROBLEMA RELACIONADO: POSTERGACION INDEFINIDA:**
- v ES POSIBLE QUE UN PROCESO SEA POSTERGADO INDEFINIDAMENTE EN TANTO QUE OTROS RECIBEN LA ATENCION DEL SISTEMA:
 - v SE TRATA DE LA **POSTERGACION INDEFINIDA:**
 - ⊕ CUANDO LOS RECURSOS SON PLANIFICADOS EN FUNCION DE PRIORIDADES, UN PROCESO DADO PUEDE ESPERAR INDEFINIDAMENTE, MIENTRAS SIGAN LLEGANDO PROCESOS DE PRIORIDADES MAYORES.
- v EN ALGUNOS SISTEMAS, LA POSTERGACION INDEFINIDA SE EVITA AL PERMITIR QUE LA PRIORIDAD DE UN PROCESO AUMENTE MIENTRAS ESPERA POR UN RECURSO:
 - v A ESTO SE LLAMA ENVEJECIMIENTO.

CONCEPTOS DE RECURSOS

- v EL S. O. ES, SOBRE TODO, UN ADMINISTRADOR DE RECURSOS.
- v LOS RECURSOS PUEDEN SER “APROPIATIVOS”, COMO LA CPU Y LA MEMORIA PRINCIPAL.
- v *LA APROPIATIVIDAD ES EXTREMADAMENTE IMPORTANTE PARA EL EXITO DE LOS SISTEMAS COMPUTACIONALES MULTIPROGRAMADOS.*
- v CIERTOS RECURSOS SON “NO APROPIATIVOS”, COMO LAS UNIDADES DE CINTA O CARTRIDGE MAGNETICOS:
 - u NO PUEDEN SACARSE DE LOS PROCESOS A LOS QUE ESTAN ASIGNADOS.
- v ALGUNOS RECURSOS:
 - u PUEDEN SER COMPARTIDOS ENTRE VARIOS PROCESOS.
 - u PUEDEN ESTAR DEDICADOS A PROCESOS INDIVIDUALES.
- v TAMBIEN SON RECURSOS COMPARTIBLES (DE USO COMPARTIDO) CIERTOS PROGRAMAS:
 - u SE CARGA UNA COPIA DEL CODIGO A MEMORIA.
 - u SE HABILITAN VARIAS COPIAS DE LAS ESTRUCTURAS DE DATOS, UNA PARA CADA USUARIO.
 - u COMO EL CODIGO PUEDE SER UTILIZADO POR VARIOS USUARIOS A LA VEZ, NO PUEDE CAMBIAR DURANTE LA EJECUCION:
 - Φ EL CODIGO QUE NO CAMBIA DURANTE LA EJECUCION SE DENOMINA *REENTRANTE*.
 - Φ EL CODIGO QUE PUEDE SER CAMBIADO, PERO SE INICIALIZA CADA VEZ QUE SE USA, SE DENOMINA *REUTILIZABLE EN SERIE*.
- v EL CODIGO REENTRANTE PUEDE SER COMPARTIDO SIMULTANEAMENTE POR VARIOS PROCESOS.
- v EL CODIGO REUTILIZABLE EN SERIE PUEDE SER USADO SOLO POR UN PROCESO A LA VEZ.
- v CUANDO SE CONSIDERAN COMPARTIDOS A DETERMINADOS RECURSOS:
 - u SE DEBE ESTABLECER SI SON UTILIZABLES POR VARIOS PROCESOS SIMULTANEAMENTE O DE A UNO POR VEZ:
 - Φ ESTOS ULTIMOS SON LOS RECURSOS QUE MAS A MENUDO ESTAN IMPLICADOS EN LOS INTERBLOQUEOS.

BLOQUEOS Y CONDICIONES NECESARIAS PARA EL BLOQUEO

- v LA SECUENCIA DE EVENTOS NECESARIOS PARA UTILIZAR UN RECURSO ES LA SIGUIENTE:
 - υ **SOLICITAR** EL RECURSO.
 - υ **UTILIZAR** EL RECURSO.
 - υ **LIBERAR** EL RECURSO.
- v SI EL RECURSO NO ESTA DISPONIBLE CUANDO SE LO SOLICITA:
 - υ EL PROCESO SOLICITANTE DEBE ESPERAR.
 - υ EN ALGUNOS S. O. EL PROCESO SE BLOQUEA AUTOMATICAMENTE Y SE DESPIERTA CUANDO DICHO RECURSO ESTA DISPONIBLE.
 - υ EN OTROS S. O. LA SOLICITUD FALLA Y EL PROCESO DEBE ESPERAR PARA LUEGO INTENTAR NUEVAMENTE.
- v UN *BLOQUEO* SE PUEDE DEFINIR FORMALMENTE COMO SIGUE:
 - υ *UN CONJUNTO DE PROCESOS SE BLOQUEA SI CADA PROCESO DEL CONJUNTO ESPERA UN EVENTO QUE SOLO PUEDE SER PROVOCADO POR OTRO PROCESO DEL CONJUNTO:*
 - ⊕ YA QUE TODOS LOS PROCESOS ESTAN ESPERANDO:
 - NINGUNO REALIZARA UN EVENTO QUE PUEDA DESPERTAR A LOS DEMAS MIEMBROS DEL CONJUNTO.
 - TODOS LOS PROCESOS ESPERARAN POR SIEMPRE.
 - ⊕ GENERALMENTE EL EVENTO QUE ESPERA CADA PROCESO ES LA LIBERACION DE CIERTO RECURSO QUE POSEE POR EL MOMENTO OTRO MIEMBRO DEL CONJUNTO:
 - CADA MIEMBRO DEL CONJUNTO DE PROCESOS BLOQUEADOS ESPERA UN RECURSO POSEIDO POR UN PROCESO BLOQUEADO.
 - NINGUNO DE LOS PROCESOS BLOQUEADOS PUEDE CONTINUAR SU EJECUCION, NI LIBERAR RECURSOS, NI PUEDE SER DESPERTADO.

BLOQUEOS Y CONDICIONES NECESARIAS PARA EL BLOQUEO

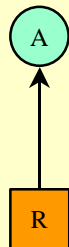
MODELACION DE BLOQUEOS

- v LAS *CONDICIONES NECESARIAS PARA EL BLOQUEO* SON (COFFMAN):
 - υ LOS PROCESOS RECLAMAN CONTROL EXCLUSIVO DE LOS RECURSOS QUE PIDEN (CONDICION DE *EXCLUSION MUTUA*).
 - υ LOS PROCESOS MANTIENEN LOS RECURSOS QUE YA LES HAN SIDO ASIGNADOS MIENTRAS ESPERAN POR RECURSOS ADICIONALES (CONDICION DE *ESPERA POR*).
 - υ LOS RECURSOS NO PUEDEN SER EXTRAIDOS DE LOS PROCESOS QUE LOS TIENEN HASTA SU COMPLETA UTILIZACION (CONDICION DE *NO APROPIATIVIDAD*).
 - υ EXISTE UNA CADENA CIRCULAR DE PROCESOS EN LA QUE C / U MANTIENE A UNO O MAS RECURSOS QUE SON REQUERIDOS POR EL SIGUIENTE PROCESO DE LA CADENA (CONDICION DE *ESPERA CIRCULAR*).
- v LA *MODELACION DE BLOQUEOS* SE PUEDE MOSTRAR MEDIANTE GRAFICAS DIRIGIDAS (HOLT).
- v LAS GRAFICAS TIENEN DOS TIPOS DE NODOS:
 - υ PROCESOS (APARECEN COMO CIRCULOS).
 - υ RECURSOS (APARECEN COMO CUADRADOS).
 - υ UN ARCO DE UN NODO DE RECURSO A UNO DE PROCESO INDICA QUE EL RECURSO:
 - Φ FUE SOLICITADO CON ANTERIORIDAD, FUE OTORGADO Y ES POSEIDO EN ESE MOMENTO POR DICHO PROCESO.
 - υ UN ARCO DE UN PROCESO A UN RECURSO INDICA QUE EL PROCESO ESTA BLOQUEADO, EN ESPERA DE ESE RECURSO.
 - υ UN CICLO EN LA GRAFICA INDICA LA EXISTENCIA DE UN BLOQUEO RELACIONADO CON LOS PROCESOS Y RECURSOS EN EL CICLO.
- v LAS *ESTRATEGIAS UTILIZADAS PARA ENFRENTAR LOS BLOQUEOS* SON:
 - υ IGNORAR TODO EL PROBLEMA.
 - υ DETECCION Y RECUPERACION.
 - υ EVITARLOS DINAMICAMENTE MEDIANTE UNA CUIDADOSA ASIGNACION DE RECURSOS.
 - υ PREVENCIÓN MEDIANTE LA NEGACION ESTRUCTURAL DE UNA DE LAS CUATRO CONDICIONES NECESARIAS.

MODELACION DE BLOQUEOS

GRAFICAS DE ASIGNACION DE RECURSOS

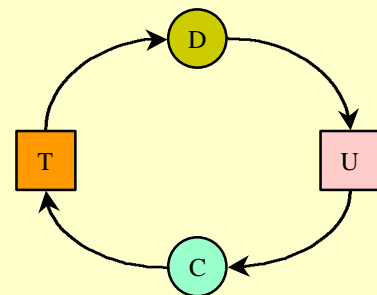
POSESION DE UN RECURSO



SOLICITUD DE UN RECURSO



BLOQUEO



EJEMPLO DE LA OCURRENCIA DE UN BLOQUEO Y LA FORMA DE EVITARLO

PROCESOS: A, B, C

RECURSOS: R, S, T

SECUENCIA DEL PROCESO A:

SOLICITUD DE R, SOLICITUD DE S, LIBERACION DE R, LIBERACION DE S.

SECUENCIA DEL PROCESO B:

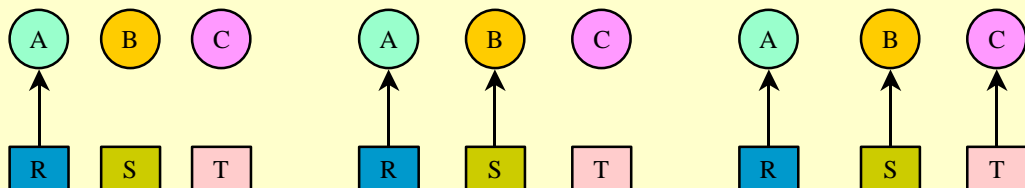
SOLICITUD DE S, SOLICITUD DE T, LIBERACION DE S, LIBERACION DE T.

SECUENCIA DEL PROCESO C:

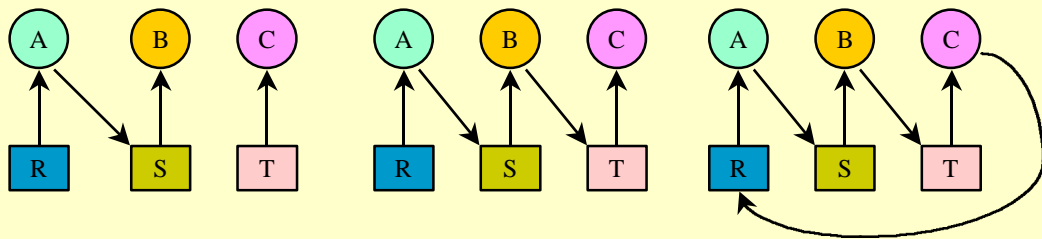
SOLICITUD DE T, SOLICITUD DE R, LIBERACION DE T, LIBERACION DE R.

SECUENCIA DE SOLICITUDES DE RECURSOS QUE CONDUCE A BLOQUEO:

A SOLICITUD R, B SOLICITUD S, C SOLICITUD T,
A SOLICITUD S, B SOLICITUD T, C SOLICITUD R, BLOQUEO.

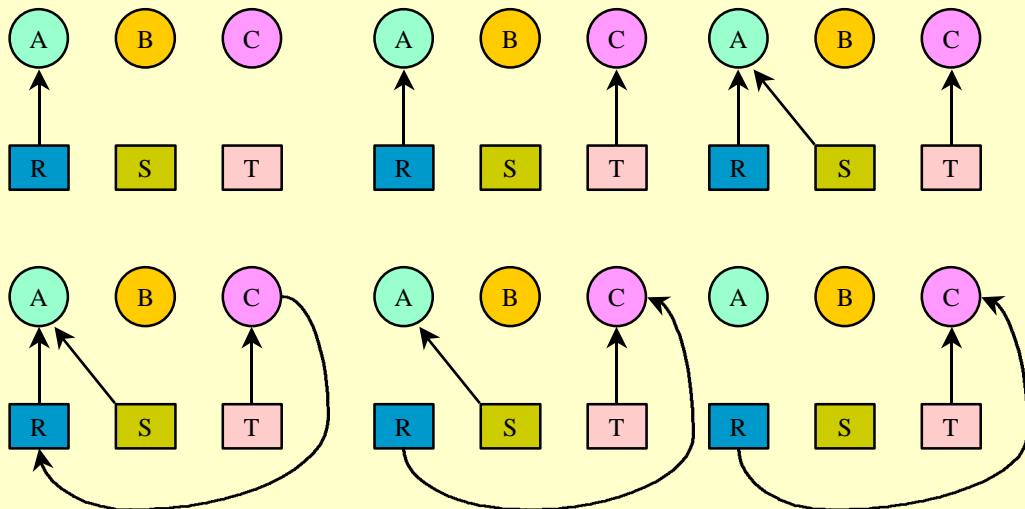


MODELACION DE BLOQUEOS



SECUENCIA DE SOLICITUDES DE RECURSOS QUE NO CONDUCE A BLOQUEO:

A SOLICITUD R, C SOLICITUD T, A SOLICITUD S,
C SOLICITUD R, A LIBERA R, A LIBERA S, NO EXISTE BLOQUEO.



AREAS PRINCIPALES EN LA INVESTIGACION DE BLOQUEOS

- v LOS PRINCIPALES ASPECTOS SON LOS SIGUIENTES:
 - u PREVENCIÓN DEL BLOQUEO.
 - u EVITACIÓN DEL BLOQUEO.
 - u DETECCIÓN DEL BLOQUEO.
 - u RECUPERACIÓN DEL BLOQUEO.
- v *PREVENCIÓN DEL BLOQUEO:*
 - u EL INTERÉS SE CENTRA EN CONDICIONAR UN SISTEMA PARA QUE ELIMINE TODA POSIBILIDAD DE QUE ESTOS SE PRODUZCAN.
 - u LOS MÉTODOS PUEDEN DAR COMO RESULTADO UNA POBRE UTILIZACIÓN DE LOS RECURSOS:
 - Φ AUN ASI SON AMPLIAMENTE UTILIZADOS.
- v *EVITACIÓN DEL BLOQUEO:*
 - u LA META ES IMPONER CONDICIONES MENOS ESTRUCTAS QUE EN LA PREVENCIÓN:
 - Φ PARA INTENTAR LOGRAR UNA MEJOR UTILIZACIÓN DE LOS RECURSOS.
 - u NO PRECONDICIONA AL SISTEMA PARA QUE EVITE TODAS LAS POSIBILIDADES DE QUE SE PRODUZCA UN BLOQUEO.
 - u PERMITEN LA APARICIÓN DEL BLOQUEO, PERO SIEMPRE QUE SE PRODUCE UNA POSIBILIDAD DE BLOQUEO, ESTE SE ESQUIVA.
- v *DETECCIÓN DEL BLOQUEO:*
 - u SE UTILIZA EN SISTEMAS QUE PERMITEN QUE ESTOS OCURRAN, YA SEA VOLUNTARIA O INVOLUNTARIAMENTE.
 - u LA META ES DETERMINAR SI HA OCURRIDO UN BLOQUEO:
 - Φ SE DEBE DETECTAR CON PRECISIÓN LOS PROCESOS Y RECURSOS IMPLICADOS EN EL BLOQUEO.
 - Φ SE PUEDE ELIMINAR EL BLOQUEO DETECTADO.
- v *RECUPERACIÓN DEL BLOQUEO:*
 - u SE UTILIZA PARA DESPEJAR BLOQUEOS DE UN SISTEMA PARA QUE:
 - Φ CONTINUE OPERANDO SIN ELLOS.
 - Φ TERMINEN LOS PROCESOS ESTANCADOS.
 - Φ SE LIBEREN LOS RECURSOS CORRESPONDIENTES A ELLOS.
 - u GENERALMENTE SE LOGRA “EXTRAYENDO” (CANCELANDO) A UNO O VARIOS DE LOS PROCESOS BLOQUEADOS:
 - Φ SE REINICIAN LUEGO DE FORMA NORMAL.

EL ALGORITMO DEL AVESTRUZ O DE OSTRICH

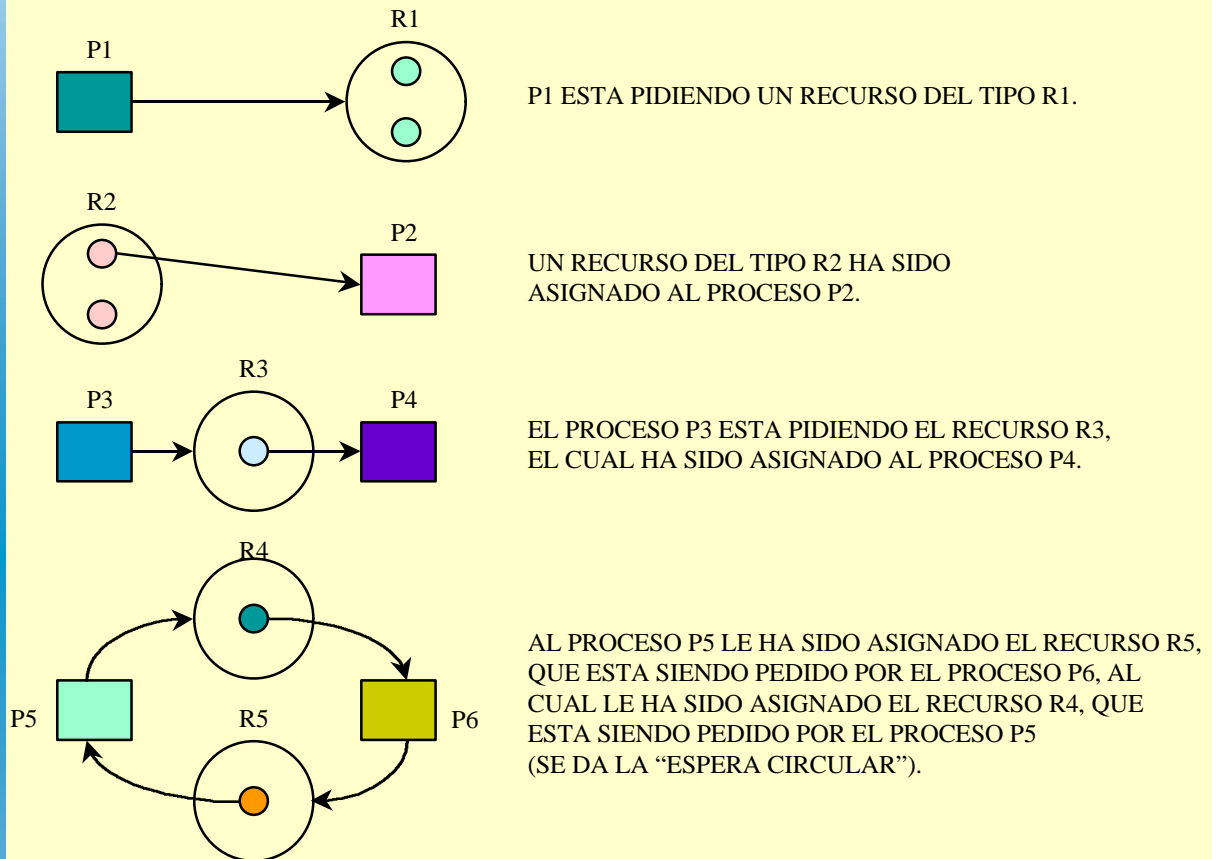
- v EL PUNTO DE VISTA MAS SIMPLE ES PRETENDER QUE NO EXISTE EL PROBLEMA.
- v ESTA ESTRATEGIA PUEDE GENERAR DISTINTAS REACCIONES:
 - u MATEMATICAMENTE ES INACEPTABLE, CONSIDERANDOSE QUE LOS BLOQUEOS DEBEN EVITARSE A TODA COSTA.
 - u DESDE LA INGENIERIA DE SOFTWARE PODRIA CONSIDERARSE CUAL ES LA FRECUENCIA ESPERADA DEL PROBLEMA, CUALES SON SUS CONSECUENCIAS ESPERADAS, CUALES SON LAS FRECUENCIAS ESPERADAS DE FALLAS DE OTRO TIPO, ETC.
- v ALGUNOS S. O. SOPORTAN POTENCIALMENTE BLOQUEOS QUE NI SIQUIERA SE DETECTAN, YA QUE SE ROMPEN AUTOMATICAMENTE.
- v LOS S. O. QUE IGNORAN EL PROBLEMA DE LOS BLOQUEOS ASUMEN LA SIGUIENTE HIPOTESIS:
 - u LA MAYORIA DE LOS USUARIOS PREFERIRIA UN BLOQUEO OCASIONAL, EN VEZ DE UNA REGLA QUE RESTRINGIERA A TODOS LOS USUARIOS EN EL USO DE LOS DISTINTOS TIPOS DE RECURSOS.
- v EL PROBLEMA ES QUE SE DEBE PAGAR UN CIERTO PRECIO PARA ENCARAR EL PROBLEMA DEL BLOQUEO:
 - u EN RESTRICCIONES PARA LOS PROCESOS.
 - u EN EL USO DE RECURSOS.
- v SE PRESENTA UNA CONTRADICCION ENTRE LA CONVENIENCIA Y LO QUE ES CORRECTO.
- v ES MUY DIFICIL ENCONTRAR TEORICAMENTE SOLUCIONES PRACTICAS DE ORDEN GENERAL APLICABLES A TODOS LOS TIPOS DE S. O.
- v UN CRITERIO DE ORDEN GENERAL UTILIZADO POR LOS S. O. QUE NO HACEN TRATAMIENTO ESPECIFICO DEL BLOQUEO CONSISTE EN:
 - u INTENTAR ACCEDER AL RECURSO COMPARTIDO.
 - u DE NO SER FACTIBLE EL ACCESO:
 - Φ ESPERAR UN TIEMPO ALEATORIO.
 - Φ REINTENTAR NUEVAMENTE.

DETECCION DE BLOQUEOS

- v EL S. O. NO INTENTA EVITAR LOS BLOQUEOS:
 - INTENTA DETECTAR CUANDO HAN OCURRIDO.
 - ACCIONA PARA RECUPERARSE DESPUES DEL HECHO.
- v LA DETECCION DEL BLOQUEO ES EL PROCESO DE:
 - DETERMINAR SI DE HECHO EXISTE O NO UN BLOQUEO.
 - IDENTIFICAR CUALES SON LOS PROCESOS Y RECURSOS IMPLICADOS EN EL BLOQUEO.
- v LOS ALGORITMOS DE DETECCION DE BLOQUEOS IMPLICAN CIERTA SOBRECARGA EN TIEMPO DE EJECUCION:
 - SURGE EL SIGUIENTE INTERROGANTE:
 - Φ ¿ COMPENSA LA SOBRECARGA IMPLICITA EN LOS ALGORITMOS DE DETECCION DE BLOQUEOS, EL AHORRO POTENCIAL DE LOCALIZARLOS Y ROMPERLOS ?.
- v **GRAFICAS DE ASIGNACION DE RECURSOS:**
- v UNA GRAFICA DIRIGIDA INDICA LAS ASIGNACIONES Y PETICIONES DE RECURSOS.
- v LOS CUADROS REPRESENTAN PROCESOS.
- v LOS CIRCULOS GRANDES INDICAN CLASES DE RECURSOS IDENTICOS.
- v LOS CIRCULOS PEQUEÑOS, DIBUJADOS DENTRO DE LOS GRANDES, REPRESENTAN EL N° DE RECURSOS IDENTICOS DENTRO DE CADA CLASE.
- v **REDUCCION DE GRAFICAS DE ASIGNACION DE RECURSOS:**
- v SI LAS PETICIONES DE RECURSOS DE UN PROCESO PUEDEN SER CONCEDIDAS:
 - SE DICE QUE UNA GRAFICA PUEDE SER *REDUCIDA* POR ESE PROCESO.
- v LA REDUCCION DE UNA GRAFICA POR UN PROCESO DETERMINADO SE MUESTRA RETIRANDO:
 - LAS FLECHAS QUE VAN DE LOS RECURSOS AL PROCESO (LOS RECURSOS ASIGNADOS AL PROCESO).
 - LAS FLECHAS QUE VAN DEL PROCESO AL RECURSO (LAS PETICIONES ACTUALES DEL PROCESO).
- v *SI UNA GRAFICA PUEDE SER REDUCIDA POR TODOS SUS PROCESOS, ENTONCES NO HAY INTERBLOQUEO.*
- v *SI UNA GRAFICA NO PUEDE SER REDUCIDA POR TODOS SUS PROCESOS, ENTONCES LOS PROCESOS “IRREDUCIBLES” CONSTITUYEN LA SERIE DE PROCESOS INTERBLOQUEADOS DE LA GRAFICA.*

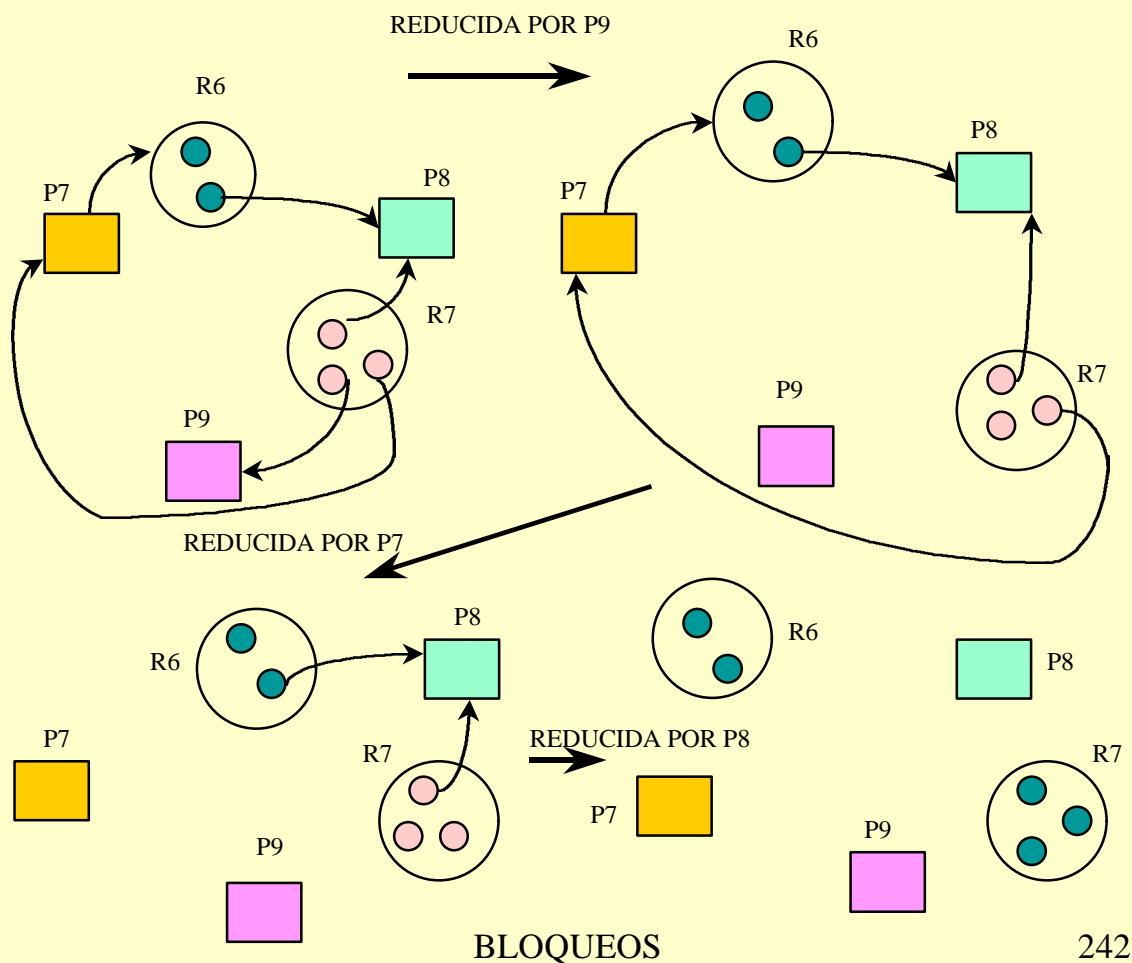
DETECCION DE BLOQUEOS

GRAFICA DE ASIGNACION Y PETICION DE RECURSOS



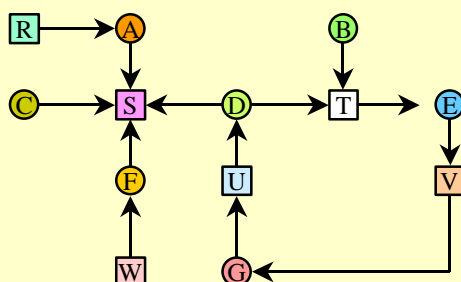
DETECCION DE BLOQUEOS

REDUCCIONES DE GRAFICAS



DETECCION DE BLOQUEOS

- v **DETECCION DE BLOQUEOS DE FORMA “UN RECURSO DE CADA TIPO”:**
- v NO SE DISPONE DE MAS DE UN OBJETO DE CADA CLASE DE RECURSO.
- v SI LA GRAFICA DE RECURSOS CONTUVIERA UNO O MAS CICLOS:
 - v EXISTIRIA UN BLOQUEO.
- v CUALQUIER PROCESO QUE FORME PARTE DE UN CICLO ESTA BLOQUEADO:
 - v SI NO EXISTEN CICLOS, EL SISTEMA NO ESTA BLOQUEADO.
- v EJEMPLO: SISTEMA CON 7 PROCESOS (“A” A “G”) Y 6 RECURSOS (“R” A “W”):
 - v LA POSESION DE LOS RECURSOS ES LA SIGUIENTE:
 - ⊕ EL PROCESO A POSEE A R Y DESEA A S.
 - ⊕ EL PROCESO B NO POSEE RECURSO ALGUNO Y DESEA A T.
 - ⊕ EL PROCESO C NO POSEE RECURSO ALGUNO Y DESEA A S.
 - ⊕ EL PROCESO D POSEE A U Y DESEA S Y A T.
 - ⊕ EL PROCESO E POSEE A T Y DESEA A V.
 - ⊕ EL PROCESO F POSEE A W Y DESEA A S.
 - ⊕ EL PROCESO G POSEE A V Y DESEA A U.
 - v LA PREGUNTA ES:
 - ⊕ ¿ESTA BLOQUEADO ESTE SISTEMA Y, EN TAL CASO, CUALES SON LOS PROCESOS BLOQUEADOS?.
 - v LA RESPUESTA SE OBTIENE MEDIANTE LA GRAFICA DE RECURSOS:
 - ⊕ SI LA GRAFICA PRESENTA UN CICLO SIGNIFICA PROCESOS BLOQUEADOS.



LOS PROCESOS D, E Y G
ESTAN BLOQUEADOS.

LOS PROCESOS A, C Y F
NO ESTAN BLOQUEADOS.

DETECCION DE BLOQUEOS

- v SE HACE NECESARIO UN ALGORITMO FORMAL PARA LA DETECCION DE BLOQUEOS QUE SE PUEDA UTILIZAR EN LOS SISTEMAS REALES.
- v EJEMPLO DE ALGORITMO APLICABLE A CADA NODO N DE LA GRAFICA:
 - u 1 - SE CONSIDERA A N COMO NODO INICIAL.
 - u 2 - SE INICIALIZAN:
 - ⊕ LA ESTRUCTURA DE DATOS L COMO UNA LISTA VACIA.
 - ⊕ TODOS LOS ARCOS COMO NO MARCADOS.
 - u 3 - SE AÑADE EL NODO ACTIVO AL FINAL DE L Y SE VERIFICA SI EL NODO APARECE EN L DOS VECES:
 - ⊕ SI APARECE DOS VECES EXISTE UN CICLO Y EL ALGORITMO TERMINA.
 - u 4 - DESDE EL NODO DADO SE VERIFICA SI EXISTEN ARCOS QUE SALGAN DE DICHO NODO Y NO ESTEN MARCADOS:
 - ⊕ EN CASO AFIRMATIVO SE VA AL PASO 5.
 - ⊕ EN CASO NEGATIVO SE VA AL PASO 6.
 - u 5 - SE ELIGE AL AZAR UN ARCO DE SALIDA NO MARCADO Y SE LE MARCA:
 - ⊕ LUEGO SE SIGUE ESTE ARCO HASTA EL NUEVO NODO ACTIVO Y SE REGRESA AL PASO 3.
 - u 6 - SE HA LLEGADO A UN PUNTO DONDE NO SE PUEDE CONTINUAR:
 - ⊕ SE REGRESA AL NODO ANTERIOR, ES DECIR AL QUE ESTABA ACTIVO ANTES DEL ACTUAL.
 - ⊕ SE SEÑALA DE NUEVO COMO NODO ACTIVO.
 - ⊕ SE PASA AL PASO 3.
 - ⊕ SI ESTE NODO ERA EL NODO INICIAL:
 - LA GRAFICA NO CONTIENE CICLOS Y EL ALGORITMO TERMINA.

DETECCION DE BLOQUEOS

- v LA APLICACION DEL ALGORITMO PRECEDENTE AL EJEMPLO ANTERIOR DE GRAFICA DIRIGIDA ES LA SIGUIENTE:
 - v SE PARTE DE R Y SE INICIALIZA L COMO LA LISTA VACIA.
 - v SE AÑADE R A LA LISTA Y SE MUEVE A LA UNICA POSIBILIDAD, A .
 - v SE AÑADE A A LA LISTA: $L=[R,A]$.
 - v SE PASA DE A A S , QUEDANDO $L=[R,A,S]$.
 - v S NO TIENE ARCOS QUE SALGAN DE EL, POR LO QUE NO SE PUEDE CONTINUAR Y SE REGRESA A A .
 - v YA QUE A NO TIENE ARCOS DE SALIDA NO MARCADOS SE REGRESA A R , FINALIZANDO LA INSPECCION DE R .
 - v SE INICIA NUEVAMENTE EL ALGORITMO PARTIENDO DE A , SIENDO L OTRA VEZ LA LISTA VACIA.
 - v LA BUSQUEDA TERMINA RAPIDAMENTE Y SE PARTE DE B .
 - v DE B SE SIGUEN LOS ARCOS DE SALIDA HASTA LLEGAR A D , SIENDO $L=[B,T,E,V,G,U,D]$.
 - v SE EFECTUA UNA ELECCION AL AZAR.
 - v SI SE ELIGE S LLEGAMOS A UN PUNTO SIN SALIDA Y DEBEMOS REGRESAR A D .
 - v LA SEGUNDA VEZ SE ELIGE T QUEDANDO $L=[B,T,E,V,G,U,D,T]$:
 - ⊕ SE HA DESCUBIERTO UN CICLO Y EL ALGORITMO SE DETIENE.
- v **DETECCION DE BLOQUEOS DE FORMA “VARIOS RECURSOS DE CADA TIPO”:**
- v SE CONSIDERA UN ALGORITMO BASADO EN MATRICES PARA LA DETECCION DE UN BLOQUEO ENTRE n PROCESOS, P_1 HASTA P_n .
- v SE CONSIDERA m EL N° DE CLASES DE RECURSOS CON:
 - v E_1 RECURSOS DE LA CLASE 1.
 - v E_2 RECURSOS DE LA CLASE 2.
 - v E_i RECURSOS DE LA CLASE i (1 MENOR O IGUAL QUE i MENOR O IGUAL QUE m).
 - v E ES EL VECTOR DE RECURSOS EXISTENTES.

DETECCION DE BLOQUEOS

- v EN TODO MOMENTO ALGUNOS DE LOS RECURSOS ESTAN ASIGNADOS:
 - u NO ESTAN DISPONIBLES.
- v SE CONSIDERA UN **VECTOR A DE RECURSOS DISPONIBLES**:
 - u A_i INDICA EL N° DE INSTANCIAS DISPONIBLES DEL RECURSO i :
 - Φ SE REFIERE A RECURSOS NO ASIGNADOS.
- v SE UTILIZAN:
 - u LA **MATRIZ C DE LA ASIGNACION ACTUAL**.
 - u LA **MATRIZ R DE SOLICITUDES**.
- v EL RENGLON i -ESIMO DE C INDICA EL N° DE INSTANCIAS DE CADA CLASE P_i POSEIDAS EN ESE MOMENTO.
- v C_{ij} ES EL N° DE INSTANCIAS DEL RECURSO j DESEADAS POR P_i .
- v CADA RECURSO ESTA ASIGNADO O DISPONIBLE:
 - u LA SUMA DE LAS INSTANCIAS DEL RECURSO j ASIGNADAS Y EL N° DE INSTANCIAS DISPONIBLES ES EL N° DE INSTANCIAS EXISTENTES DE ESA CLASE DE RECURSO.
- v EL ALGORITMO DE DETECCION DE BLOQUEOS SE BASA EN LA COMPARACION DE VECTORES:
 - u DEFINIMOS QUE A ES MENOR O IGUAL QUE B SI Y SOLO SI A_i ES MENOR O IGUAL QUE B_i PARA i ENTRE 0 Y m , AMBOS INCLUSIVE.
- v LOS PROCESOS NO ESTAN MARCADOS AL PRINCIPIO.
- v AL AVANZAR EL ALGORITMO LOS PROCESOS SE MARCARAN:
 - u ESTO INDICA QUE PUEDEN TERMINAR SU LABOR, YA QUE NO ESTAN BLOQUEADOS.
 - u AL CONCLUIR EL ALGORITMO SE SABE QUE LOS PROCESOS NO MARCADOS ESTARAN BLOQUEADOS.
- v LOS PASOS BASICOS DEL ALGORITMO DE DETECCION DE BLOQUEOS SON LOS SIGUIENTES:
 - u 1 - SE BUSCA UN PROCESO NO MARCADO P_i , PARA EL CUAL EL i -ESIMO RENGLON DE R SEA MENOR QUE A .
 - u 2 - SI SE ENCUENTRA TAL PROCESO, SE SUMA EL i -ESIMO RENGLON DE C A A , SE MARCA EL PROCESO Y SE REGRESA AL PASO 1.
 - u SI NO EXISTE TAL PROCESO, EL ALGORITMO TERMINA.

DETECCION DE BLOQUEOS

ESTRUCTURAS DE DATOS NECESARIAS PARA EL ALGORITMO DE DETECCION DE BLOQUEOS

RECURSOS EN EXISTENCIA: $(E_1, E_2, E_3, \dots, E_m)$

RECURSOS DISPONIBLES: $(A_1, A_2, A_3, \dots, A_m)$

MATRIZ DE ASIGNACION ACTUAL:
$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \dots & C_{1m} \\ C_{21} & C_{22} & C_{23} & \dots & C_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & C_{n3} & \dots & C_{nm} \end{bmatrix}$$
 EL RENGLON n ES LA ASIGNACION ACTUAL PARA EL PROCESO n .

MATRIZ DE SOLICITUDES:
$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1m} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{n1} & R_{n2} & R_{n3} & \dots & R_{nm} \end{bmatrix}$$
 EL RENGLON 2 ES LO QUE NECESITA EL PROCESO 2.

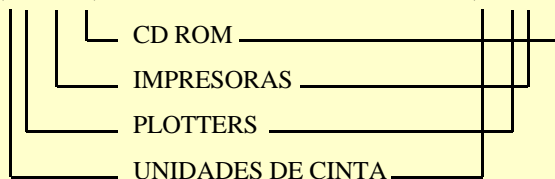
UN EJEMPLO DEL ALGORITMO DE DETECCION DE BLOQUEOS

RECURSOS EN EXISTENCIA:

$E = (4 \ 2 \ 3 \ 1)$

RECURSOS DISPONIBLES:

$A = (2 \ 1 \ 0 \ 0)$



MATRIZ DE ASIGNACION ACTUAL:

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix}$$

MATRIZ DE SOLICITUDES:

$$R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

DETECCION DE BLOQUEOS

- v EN EL EJEMPLO TENEMOS 3 PROCESOS Y 4 CLASES DE RECURSOS.
- v EL PROCESO 1 TIENE 1 IMPRESORA.
- v EL PROCESO 2 TIENE 2 UNIDADES DE CINTA Y 1 UNIDAD DE CD ROM.
- v EL PROCESO 3 TIENE 1 PLOTTER Y 2 IMPRESORAS.
- v LA MATRIZ R INDICE LAS NECESIDADES DE RECURSOS ADICIONALES.
- v EL ALGORITMO DE DETECCION DE BLOQUEOS BUSCA UN PROCESO CUYA SOLICITUD DE UN RECURSO PUEDA SER SATISFECHA:
 - u EL PROCESO 1 NO SE PUEDE SATISFACER POR NO DISPONER DE UNA UNIDAD DE CD ROM.
 - u EL PROCESO 2 NO SE PUEDE SATISFACER POR NO DISPONER DE UNA IMPRESORA.
 - u EL PROCESO 3 SI SE PUEDE SATISFACER, POR LO QUE SE EJECUTA, REGRESANDO EN CIERTO MOMENTO SUS RECURSOS, LO QUE RESULTA EN: $A = (2 \ 2 \ 2 \ 0)$.
- v SE EJECUTA EL PROCESO 2, EL CUAL REGRESA SUS RECURSOS, OBTENIENDOSE: $A = (4 \ 2 \ 2 \ 1)$.
- v SE EJECUTA EL PROCESO RESTANTE:
 - u NO EXISTE BLOQUEO EN EL SISTEMA.
- v SI SE CONSIDERA LA SIGUIENTE VARIANTE:
 - u EL PROCESO 2 NECESITA 1 UNIDAD DE CD ROM, LAS 2 UNIDADES DE CINTA Y EL PLOTTER.
 - u NO SE PUEDEN SATISFACER LAS 3 SOLICITUDES Y TODO EL SISTEMA SE BLOQUEA.
- v **CUANDO BUSCAR LOS BLOQUEOS:**
- v UNA POSIBILIDAD ES CADA VEZ QUE SE SOLICITA UN RECURSO:
 - u ESTO PODRIA SOBRECARGAR AL SISTEMA.
- v OTRA POSIBILIDAD ES VERIFICAR CADA k MINUTOS.
- v OTRO CRITERIO ES VERIFICAR CUANDO EL USO DE LA CPU BAJE DE CIERTO VALOR FIJO:
 - u SI SE BLOQUEAN SUFICIENTES PROCESOS:
 - Φ EXISTIRAN POCOS PROCESOS EN EJECUCION.
 - Φ LA CPU ESTARA INACTIVA CON MAS FRECUENCIA.

RECUPERACION DE BLOQUEOS

- v PARA ROMPER EL BLOQUEO DE UN SISTEMA HAY QUE ANULAR UNA O MAS DE LAS CONDICIONES NECESARIAS PARA EL BLOQUEO.
- v NORMALMENTE, VARIOS PROCESOS PERDERAN ALGO O TODO LO REALIZADO HASTA EL MOMENTO.
- v LOS PRINCIPALES FACTORES QUE DIFICULTAN LA RECUPERACION DEL BLOQUEO SON LOS SIGUIENTES:
 - u PUEDE NO ESTAR CLARO SI EL SISTEMA SE HA BLOQUEADO O NO.
 - u MUCHOS SISTEMAS TIENEN LIMITACIONES PARA SUSPENDER UN PROCESO POR TIEMPO INDEFINIDO Y REANUDARLO MAS TARDE:
 - Φ EJ.: LOS PROCESOS DE TIEMPO REAL, QUE DEBEN FUNCIONAR CONTINUAMENTE, NO SON FACILES DE SUSPENDER Y REANUDAR.
 - u LOS PROCEDIMIENTOS DE SUSPENSION / REANUDACION IMPLICAN UNA SOBRECARGA CONSIDERABLE.
 - u LA SOBRECARGA DE RECUPERACION ESTA EN FUNCION DE LA MAGNITUD DEL BLOQUEO (ALGUNOS, DECENAS O CENTENAS DE PROCESOS INVOLUCRADOS).
- v GENERALMENTE LA RECUPERACION SUELE REALIZARSE:
 - u RETIRANDO FORZOSAMENTE (CANCELANDO) A UN PROCESO.
 - u RECLAMANDO SUS RECURSOS.
 - u PERMITIENDO QUE LOS PROCESOS RESTANTES PUEDAN FINALIZAR.
- v LOS PROCESOS PUEDEN SER RETIRADOS (CANCELADOS) DE ACUERDO A UN ORDEN DE PRIORIDADES, EXISTIENDO LAS SIGUIENTES DIFICULTADES:
 - u PUEDEN NO EXISTIR LAS PRIORIDADES DE LOS PROCESOS BLOQUEADOS.
 - u LAS PRIORIDADES INSTANTANEAS (EN UN MOMENTO DADO), PUEDEN SER INCORRECTAS O CONFUSAS DEBIDO A CONSIDERACIONES ESPECIALES:
 - Φ EJ.: PROCESOS DE BAJA PRIORIDAD QUE TIENEN PRIORIDAD ALTA MOMENTANEAMENTE DEBIDO A UN TIEMPO TOPE INMINENTE.
 - u LA DECISION OPTIMA PUEDE REQUERIR UN GRAN ESFUERZO.

RECUPERACION DE BLOQUEOS

- v ALGUNAS **FORMAS DE RECUPERACION** ANTE BLOQUEOS SON:
 - u RECUPERACION MEDIANTE LA APROPIACION.
 - u RECUPERACION MEDIANTE ROLLBACK.
 - u RECUPERACION MEDIANTE LA ELIMINACION DE PROCESOS.
- v **RECUPERACION MEDIANTE LA APROPIACION.**
- v EN CIERTOS CASOS PODRIA SER POSIBLE TOMAR UN RECURSO TEMPORALMENTE DE SU POSEEDOR Y DARSELO A OTRO PROCESO:
 - u EJ.:
 - Φ RETIRAR UNA IMPRESORA DE UN PROCESO PARA DEDICARLA A OTRO PROCESO.
 - Φ RETOMAR LUEGO EL PRIMER PROCESO REASIGNANDOLA AL MISMO.
- v LA RECUPERACION DE RECURSOS DE ESTA FORMA DEPENDE EN GRAN MEDIDA DE LA NATURALEZA DEL RECURSO.
- v LA ELECCION DEL PROCESO A SUSPENDER DEPENDE MUCHO:
 - u DE CUALES PROCESOS POSEEN RECURSOS QUE PUEDEN SER TOMADOS CON FACILIDAD.
 - u DE LAS POSIBILIDADES DE RECUPERACION LUEGO DE LA APROPIACION.
- v **RECUPERACION MEDIANTE ROLLBACK.**
- v EN LOS S. O. DONDE ES POSIBLE QUE OCURRAN BLOQUEOS SE PUEDE HACER QUE LOS PROCESOS SEAN **VERIFICADOS** PERIODICAMENTE:
 - u SU ESTADO SE GRABA EN UN ARCHIVO DE MODO QUE PUEDA VOLVER A INICIAR MAS TARDE.
 - u **EL PUNTO DE VERIFICACION O DE CONTROL** CONTIENE:
 - Φ LA IMAGEN DE LA MEMORIA.
 - Φ EL ESTADO DE LOS RECURSOS:
 - DETALLE DE LOS RECURSOS ASIGNADOS AL PROCESO EN ESE INSTANTE.
 - u LOS PUNTOS DE VERIFICACION GRABADOS DURANTE UN PROCESO SE MANTIENEN SIN SER REGRABADOS.
- v AL DETECTARSE UN BLOQUEO ES FACIL VER CUALES SON LOS RECURSOS NECESARIOS.

RECUPERACION DE BLOQUEOS

- v PARA LA RECUPERACION:
 - u UN PROCESO QUE POSEE UN RECURSO NECESARIO REGRESA HASTA CIERTO INSTANTE EN EL TIEMPO ANTERIOR A LA ADQUISICION:
 - ⊕ INICIALIZA ALGUNO DE SUS ANTERIORES PUNTOS DE VERIFICACION.
 - ⊕ EL PROCESO REGRESA A UN MOMENTO ANTERIOR EN EL QUE NO POSEIA EL RECURSO.
 - ⊕ EL RECURSO SE ASIGNA AHORA A UNO DE LOS PROCESOS BLOQUEADOS.
 - ⊕ SI EL PROCESO QUE VOLVIO A INICIAR INTENTA ADQUIRIR DE NUEVO EL RECURSO:
 - TENDRA QUE ESPERAR HASTA QUE ESTE DISPONIBLE.
- v **RECUPERACION MEDIANTE LA ELIMINACION DE PROCESOS.**
- v ES LA FORMA MAS SENCILLA DE ROMPER UN BLOQUEO.
- v UNA POSIBILIDAD ES ELIMINAR UN PROCESO DEL CICLO:
 - u SI EL BLOQUEO NO SE ROMPE, SE PUEDE INTENTAR CON OTRO PROCESO DEL CICLO, HASTA ROMPER DICHO CICLO.
- v OTRA POSIBILIDAD ES ELIMINAR UN PROCESO QUE NO ESTE EN EL CICLO, PARA PODER LIBERAR SUS RECURSOS:
 - u DEBE ELEGIRSE UN PROCESO QUE POSEA RECURSOS NECESARIOS POR ALGUN PROCESO DEL CICLO.
- v SIEMPRE QUE SEA POSIBLE, ES MEJOR ELIMINAR UN PROCESO QUE PUEDA VOLVER A INICIAR SU EJECUCION SIN EFECTOS DAÑINOS:
 - u ES PREFERIBLE ELIMINAR UN PROCESO DE COMPILACION QUE UN PROCESO DE ACTUALIZACION DE UNA BASE DE DATOS:
 - ⊕ LA COMPILACION SE PUEDE REPETIR SIN PROBLEMAS.
 - ⊕ LA ACTUALIZACION DE UNA BASE DE DATOS NO SIEMPRE SE PUEDE REPETIR DIRECTAMENTE.

EVASION DE BLOQUEOS

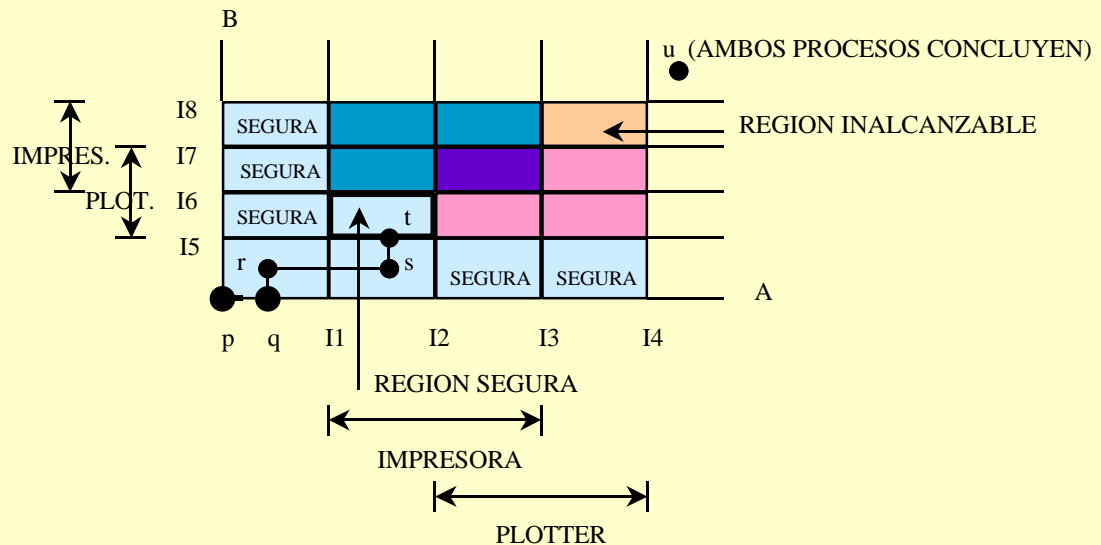
- v EN ESTE ANALISIS SE SUPONE IMPLICITAMENTE QUE SI UN PROCESO SOLICITA RECURSOS, LOS SOLICITA TODOS AL MISMO TIEMPO:
 - υ EN LA MAYORIA DE LOS SISTEMAS LOS RECURSOS SE SOLICITAN UNO A LA VEZ.
 - υ EL S. O. DEBE PODER:
 - Φ DECIDIR SI EL OTORGAMIENTO DE UN RECURSO ES SEGURO O NO.
 - Φ ASIGNARLO SOLO EN CASO DE QUE SEA SEGURO.
- v EL OBJETIVO ES EVITAR EL BLOQUEO HACIENDO LA ELECCION CORRECTA TODO EL TIEMPO:
 - υ PARA EVITAR LOS BLOQUEOS SE REQUIERE DE CIERTA INFORMACION DE ANTEMANO.
- v **TRAYECTORIAS DE RECURSOS.**
- v LOS PRINCIPALES ALGORITMOS PARA EVITAR LOS BLOQUEOS SE BASAN EN EL CONCEPTO DE **ESTADOS SEGUROS**.
- v EL EJEMPLO DE MODELO GRAFICO UTILIZADO INDICA LO SIGUIENTE:
 - υ ES VALIDO PARA DOS PROCESOS Y DOS RECURSOS.
 - υ EL EJE HORIZONTAL REPRESENTA EL N° DE INSTRUCCIONES EJECUTADAS POR EL PROCESO A.
 - υ EL EJE VERTICAL REPRESENTA EL N° DE INSTRUCCIONES EJECUTADAS POR EL PROCESO B.
 - υ EN *I1* A SOLICITA UNA IMPRESORA Y EN *I2* NECESITA UN PLOTTER.
 - υ EN *I3* E *I4* SE LIBERAN LA IMPRESORA Y EL PLOTTER.
 - υ EL PROCESO B NECESITA EL PLOTTER DESDE *I5* HASTA *I7* Y LA IMPRESORA DESDE *I6* HASTA *I8*.
 - υ CADA PUNTO DEL DIAGRAMA REPRESENTA UN ESTADO CONJUNTO DE LOS DOS PROCESOS.
 - υ EL ESTADO INICIAL ES *p*, SIN QUE LOS PROCESOS HAYAN EJECUTADO INSTRUCCION ALGUNA.
 - υ SI EL PLANIFICADOR DEL S. O. ELIGE A SE PASA A *q*, EN DONDE A HA EJECUTADO INSTRUCCIONES PERO NO B.
 - υ EN *q* LA TRAYECTORIA SE VUELVE VERTICAL, YA QUE EL PLANIFICADOR HA ELEGIDO EJECUTAR B.

EVASION DE BLOQUEOS

- υ CON UN MONOPROCESADOR TODAS LAS TRAYECTORIAS SERAN HORIZONTALES O VERTICALES (NO DIAGONALES).
- υ CUANDO A CRUZA LA LINEA II EN LA TRAYECTORIA DE r A s :
 - ϕ SOLICITA Y SE LE OTORGA LA IMPRESORA.
- υ CUANDO B ALCANZA EL PUNTO t , SOLICITA EL PLOTTER.
- υ LA REGION DELIMITADA POR II , $I3$, $I6$ E $I8$ REPRESENTA QUE AMBOS PROCESOS POSEEN LA IMPRESORA:
 - ϕ ESTO ES IMPOSIBLE Y LA REGLA DE EXCLUSION MUTUA IMPIDE LA ENTRADA A ESTA REGION.
- υ LA REGION DELIMITADA POR $I2$, $I4$, $I5$ E $I7$ REPRESENTA QUE AMBOS PROCESOS POSEEN EL PLOTTER, LO QUE ES IMPOSIBLE.
- υ SI EL SISTEMA INGRESARA A LA REGION DELIMITADA POR II , $I2$, $I5$ E $I6$ SE BLOQUEARA EN LA INTERSECCION DE $I2$ E $I6$:
 - ϕ ACA, A SOLICITA EL PLOTTER Y B LA IMPRESORA, QUE YA ESTAN ASIGNADOS.
 - ϕ TODA LA REGION NO ES SEGURA Y NO HAY QUE ENTRAR A ELLA:
 - EN t , LO UNICO SEGURO ES EJECUTAR A HASTA LLEGAR A $I4$.
 - LUEGO SE PUEDE UTILIZAR CUALQUIER TRAYECTORIA HASTA u .
- υ EN t , B SOLICITA UN RECURSO:
 - ϕ EL S. O. DEBE DECIDIR SI LO OTORGA O NO.
 - ϕ SI LO OTORGA, EL SISTEMA ENTRARA A UNA REGION INSEGURA Y SE BLOQUEARA EN ALGUN MOMENTO.
 - ϕ PARA EVITAR EL BLOQUEO, HAY QUE SUSPENDER A B HASTA QUE A HAYA SOLICITADO Y LIBERADO EL PLOTTER.

EVASION DE BLOQUEOS

TRAYECTORIAS DE RECURSOS DE DOS PROCESOS



- v **ESTADOS SEGUROS E INSEGUROS.**
- v UN ESTADO ACTUAL ESTA CONFORMADO POR E , A , C Y R :
 - v E : VECTOR DE RECURSOS EN EXISTENCIA.
 - v A : VECTOR DE RECURSOS DISPONIBLES.
 - v C : MATRIZ DE ASIGNACION ACTUAL.
 - v R : MATRIZ DE SOLICITUDES.
- v UN ESTADO ES SEGURO SI:
 - v NO ESTA BLOQUEADO.
 - v EXISTE UNA FORMA DE SATISFACER TODAS LAS SOLICITUDES PENDIENTES, MEDIANTE LA EJECUCION DE LOS PROCESOS EN CIERTO ORDEN.
- v EJEMPLO CON UN RECURSO PARA DEMOSTRAR QUE EL ESTADO EN (a) ES SEGURO:
 - v EL ESTADO ES SEGURO YA QUE EXISTE UNA SUCESION DE ASIGNACIONES QUE PERMITEN TERMINAR A TODOS LOS PROCESOS:
 - LA SUCESION DE ASIGNACIONES ES LA SIGUIENTE:

EVASION DE BLOQUEOS

T M	T M	T M	T M	T M	T: TIENE
A 3 9	A 3 9	A 3 9	A 3 9	A 3 9	M: MAXIMO
B 2 4	B 4 4	B 0 -	B 0 -	B 0 -	L: LIBRE
C 2 7	C 2 7	C 2 7	C 7 7	C 0 -	
L: 3	L: 1	L: 5	L: 0	L: 7	
(a)	(b)	(c)	(d)	(e)	

- v EJEMPLO CON UN RECURSO PARA MOSTRAR UN ESTADO INSEGURO:
 - v NO SE PUEDE GARANTIZAR QUE TERMINEN LOS TRES PROCESOS.
 - v SI EL PROCESO A PIDE Y SE LE OTORGA UNA UNIDAD:
 - ❖ PUEDE PRODUCIRSE UN BLOQUEO DE TRES VIAS SI C / U DE LOS PROCESOS NECESITA AL MENOS OTRA UNIDAD DEL RECURSO ANTES DE LIBERAR NINGUNA.
- v *UN ESTADO INSEGURO:*
 - v *NO IMPLICA LA EXISTENCIA, NI SIQUIERA EVENTUAL, DE BLOQUEO.*
 - v *SI IMPLICA QUE ALGUNA SECUENCIA INFORTUNADA DE EVENTOS DE COMO RESULTADO UN BLOQUEO.*
- v LA DIFERENCIA ENTRE ESTADO SEGURO E INSEGURO ES QUE:
 - v A PARTIR DE UN ESTADO SEGURO, EL SISTEMA PUEDE GARANTIZAR LA CONCLUSION DE TODOS LOS PROCESOS.
 - v A PARTIR DE UN ESTADO INSEGURO, NO EXISTE TAL GARANTIA.

T M	T: TIENE
A 8 10	M: MAXIMO
B 2 5	L: LIBRE
C 1 3	
L: 1	

EVASION DE BLOQUEOS

T	M	T	M	
A	1 4	A	1 4	T: TIENE
B	4 6	B	4 6	M: MAXIMO
C	5 8	C	6 8	L: LIBRE
L:	2	L:	1	

- v EJEMPLO DE UNA TRANSICION DE ESTADO SEGURO A ESTADO INSEGURO:
 - v DADO UN ESTADO ACTUAL SEGURO, ELLO NO IMPLICA QUE VAYAN A SER SEGUROS TODOS LOS ESTADOS FUTUROS.
- v **EL ALGORITMO DEL BANQUERO (DE DIJKSTRA) PARA SOLO UN RECURSO.**
- v ES UN ALGORITMO DE PLANIFICACION QUE PUEDE EVITAR LOS BLOQUEOS.
- v EN LA ANALOGIA:
 - v LOS CLIENTES SON LOS PROCESOS, LAS UNIDADES DE CREDITO SON LOS RECURSOS DEL SISTEMA Y EL BANQUERO ES EL S. O.
 - v EL BANQUERO SABE QUE NO TODOS LOS CLIENTES NECESITARAN SU CREDITO MAXIMO OTORGADO EN FORMA INMEDIATA:
 - ⊕ RESERVA MENOS UNIDADES (RECURSOS) DE LAS TOTALES NECESARIAS PARA DAR SERVICIO A LOS CLIENTES.
- v UN ESTADO INSEGURO *NO TIENE* QUE LLEVAR A UN BLOQUEO.
- v EL ALGORITMO DEL BANQUERO CONSISTE EN:
 - v ESTUDIAR CADA SOLICITUD AL OCURRIR ESTA.
 - v VER SI SU OTORGAMIENTO CONDUCE A UN ESTADO SEGURO:
 - ⊕ EN CASO POSITIVO, SE OTORGA LA SOLICITUD.
 - ⊕ EN CASO NEGATIVO, SE LA POSPONE.
 - v PARA VER SI UN ESTADO ES SEGURO:
 - ⊕ VERIFICA SI TIENE LOS RECURSOS SUFICIENTES PARA SATISFACER A OTRO CLIENTE:
 - EN CASO AFIRMATIVO, SE SUPONE QUE LOS PRESTAMOS SE PAGARAN.
 - SE VERIFICA AL SIGUIENTE CLIENTE CERCANO AL LIMITE Y ASI SUCESIVAMENTE.
 - ⊕ SI EN CIERTO MOMENTO SE VUELVEN A PAGAR TODOS LOS CREDITOS, EL ESTADO ES SEGURO Y LA SOLICITUD ORIGINAL DEBE SER APROBADA.

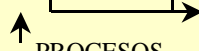
EVASION DE BLOQUEOS

- v **EL ALGORITMO DEL BANQUERO (DE DIJKSTRA) PARA VARIOS RECURSOS.**
- v ACA TAMBIEN LOS PROCESOS DEBEN ESTABLECER SUS NECESIDADES TOTALES DE RECURSOS ANTES DE SU EJECUCION:
 - v DADA UNA MATRIZ DE RECURSOS ASIGNADOS, EL S. O. DEBE PODER CALCULAR EN CUALQUIER MOMENTO LA MATRIZ DE RECURSOS NECESARIOS.
- v SE DISPONE DE:
 - v E : VECTOR DE RECURSOS EXISTENTES.
 - v P : VECTOR DE RECURSOS POSEIDOS.
 - v A : VECTOR DE RECURSOS DISPONIBLES.
- v EL ALGORITMO PARA DETERMINAR SI UN ESTADO ES SEGURO ES EL SIGUIENTE:
 - v 1. SE BUSCA UN RENGLON R CUYAS NECESIDADES DE RECURSOS NO SATISFECHAS SEAN MENORES O IGUALES QUE A :
 - ⊕ SI NO EXISTE TAL RENGLON, EL SISTEMA SE BLOQUEARA EN ALGUN MOMENTO:
 - NINGUN PROCESO PODRA CONCLUIRSE.
 - v 2. SUPONGAMOS QUE EL PROCESO DEL RENGLON ELEGIDO SOLICITA TODOS LOS RECURSOS QUE NECESITA Y CONCLUYE:
 - ⊕ SE SEÑALA EL PROCESO COMO CONCLUIDO Y SE AÑADEN SUS RECURSOS AL VECTOR A .
 - v 3. SE REPITEN LOS PASOS 1 Y 2:
 - ⊕ HASTA QUE TODOS LOS PROCESOS QUEDEN SEÑALADOS COMO CONCLUIDOS, EN CUYO CASO, EL ESTADO INICIAL ERA SEGURO, O
 - ⊕ HASTA QUE OCURRA UN BLOQUEO, EN CUYO CASO, NO LO ERA.

EL ALGORITMO DEL BANQUERO CON VARIOS RECURSOS

RECURSOS ASIGNADOS

A 3 0 1 1
 B 0 1 0 0
 C 1 1 1 0
 D 1 1 0 1
 E 0 0 0 0



RECURSOS NECESARIOS

A 1 1 0 0
 B 0 1 1 2
 C 3 1 0 0
 D 0 0 1 0
 E 2 1 1 0

$E = (6 \ 3 \ 4 \ 2)$
 $P = (5 \ 3 \ 2 \ 2)$
 $A = (1 \ 0 \ 2 \ 0)$

EVASION DE BLOQUEOS

- v **ASIGNACION DE RECURSOS POR EL ALGORITMO DEL BANQUERO.**
- v SE PERMITEN LAS CONDICIONES DE “EXCLUSION MUTUA”, “ESPERA POR” Y “NO APROPIATIVIDAD”.
- v LOS PROCESOS RECLAMAN USO EXCLUSIVO DE LOS RECURSOS QUE REQUIEREN.
- v LOS PROCESOS MANTIENEN LOS RECURSOS MIENTRAS PIDEN Y ESPERAN POR OTROS RECURSOS ADICIONALES:
 - v NO PUEDEN APROPIARSE DE UN PROCESO QUE MANTENGA ESOS RECURSOS.
- v LAS PETICIONES SON DE UN RECURSO A LA VEZ.
- v EL S. O. PUEDE CONCEDER O NEGAR C / U DE LAS PETICIONES.
 - v SI SE NIEGA UNA PETICION:
 - ⊕ EL PROCESO RETIENE LOS RECURSOS QUE YA TIENE ASIGNADOS.
 - ⊕ ESPERA UN TIEMPO FINITO HASTA QUE LE SEA ATENDIDA LA PETICION.
- v *EL S. O. CONCEDE PETICIONES QUE DEN COMO RESULTADO SOLO ESTADOS SEGUROS.*
- v DADO QUE EL SISTEMA SE MANTIENE SIEMPRE EN ESTADO SEGURO, TODAS LAS PETICIONES SERAN ATENDIDA EN UN TIEMPO FINITO.
- v **DEBILIDADES DEL ALGORITMO DEL BANQUERO.**
- v REQUIERE QUE EXISTA UN N° FIJO DE RECURSOS ASIGNABLES, PERO GENERALMENTE NO SE PUEDE CONTAR CON QUE EL N° DE RECURSOS SE MANTENGA SIEMPRE CONSTANTE.
- v REQUIERE QUE LA POBLACION DE USUARIOS SE MANTENGA CONSTANTE, LO CUAL ES IRRAZONABLE.
- v REQUIERE QUE EL S. O. GARANTICE QUE TODAS LAS PETICIONES SERAN CONCEDIDAS EN UN TIEMPO FINITO, PERO EN LA REALIDAD SE REQUIEREN MAYORES GARANTIAS.
- v REQUIERE QUE LOS PROCESOS REINTEGREN LOS RECURSOS EN UN TIEMPO FINITO, PERO EN LA REALIDAD SE REQUIEREN MAYORES GARANTIAS.
- v REQUIERE QUE LOS PROCESOS INDIQUEN SUS NECESIDADES MAXIMAS DE RECURSOS POR ADELANTADO, LO CUAL GENERALMENTE NO OCURRE.
- v GENERALMENTE NO ES UTILIZADO EN S. O. REALES.

PREVENCION DE BLOQUEOS

- v SI SE PUEDE GARANTIZAR QUE AL MENOS UNA DE LAS CUATRO CONDICIONES DE COFFMAN PARA EL BLOQUEO NUNCA SE SATISFACE:
 - u LOS BLOQUEOS SERAN IMPOSIBLES POR RAZONES ESTRUCTURALES (ENUNCIADO DE HAVENDER).
- v HAVENDER SUGIRIO LAS SIGUIENTES ESTRATEGIAS PARA EVITAR VARIAS DE LAS CONDICIONES DE BLOQUEO:
 - u CADA PROCESO:
 - Φ DEBERA PEDIR TODOS SUS RECURSOS REQUERIDOS DE UNA SOLA VEZ.
 - Φ NO PODRA PROCEDER HASTA QUE LE HAYAN SIDO ASIGNADOS.
 - u SI A UN PROCESO QUE MANTIENE CIERTOS RECURSOS SE LE NIEGA UNA NUEVA PETICION, ESTE PROCESO DEBERA:
 - Φ LIBERAR SUS RECURSOS ORIGINALES.
 - Φ EN CASO NECESARIO, PEDIRLOS DE NUEVO JUNTO CON LOS RECURSOS ADICIONALES.
 - u SE IMPONDRA LA ORDENACION LINEAL DE LOS TIPOS DE RECURSOS EN TODOS LOS PROCESOS:
 - Φ SI A UN PROCESO LE HAN SIDO ASIGNADOS RECURSOS DE UN TIPO DADO, EN LO SUCESIVO SOLO PODRA PEDIR AQUELLOS RECURSOS DE LOS TIPOS QUE SIGUEN EN EL ORDENAMIENTO.
- v HAVENDER NO PRESENTA UNA ESTRATEGIA CONTRA EL USO EXCLUSIVO DE RECURSOS POR PARTE DE LOS PROCESOS:
 - u SE DESEA PERMITIR EL USO DE RECURSOS DEDICADOS.
- v **PREVENCION DE LA CONDICION DE EXCLUSION MUTUA.**
- v SI NINGUN RECURSO SE ASIGNARA DE MANERA EXCLUSIVA A UN SOLO PROCESO, NUNCA TENDRIAMOS BLOQUEOS:
 - u ESTO ES IMPOSIBLE DE APLICAR, EN ESPECIAL EN RELACION A CIERTOS TIPOS DE RECURSOS, QUE EN UN MOMENTO DADO NO PUEDEN SER COMPARTIDOS (EJ.: IMPRESORAS).
- v SE DEBE:
 - u EVITAR LA ASIGNACION DE UN RECURSO CUANDO NO SEA ABSOLUTAMENTE NECESARIO.
 - u INTENTAR ASEGURARSE DE QUE LOS MENOS PROCESOS POSIBLES PUEDAN PEDIR EL RECURSO.

PREVENCION DE BLOQUEOS

- v **PREVENCION DE LA CONDICION “DETENERSE Y ESPERAR” O “ESPERA POR”.**
- v SI SE PUEDE EVITAR QUE LOS PROCESOS QUE CONSERVAN RECURSOS ESPEREN MAS RECURSOS, SE PUEDEN ELIMINAR LOS BLOQUEOS.
- v UNA FORMA ES EXIGIR A TODOS LOS PROCESOS QUE SOLICITEN TODOS LOS RECURSOS ANTES DE INICIAR SU EJECUCION:
 - u SI NO PUEDE DISPONER DE TODOS LOS RECURSOS, DEBERA ESPERAR, PERO SIN RETENER RECURSOS AFECTADOS.
- v UN PROBLEMA ES QUE MUCHOS PROCESOS NO SABEN EL N° DE RECURSOS NECESARIOS HASTA INICIAR SU EJECUCION.
- v OTRO PROBLEMA ES QUE PUEDE SIGNIFICAR DESPERDICIO DE RECURSOS:
 - u TODOS LOS RECURSOS NECESARIOS PARA UN PROCESO ESTAN AFECTADOS AL MISMO DESDE SU INICIO HASTA SU FINALIZACION.
- v OTRO CRITERIO APLICABLE CONSISTE EN:
 - u EXIGIR A UN PROCESO QUE SOLICITA UN RECURSO QUE LIBERE EN FORMA TEMPORAL LOS DEMAS RECURSOS QUE MANTIENE EN ESE MOMENTO.
 - u HACER QUE EL PROCESO INTENTE LUEGO RECUPERAR TODO AL MISMO TIEMPO.
- v **PREVENCION DE LA CONDICION DE “NO APROPIACION”.**
- v UNA DE LAS ESTRATEGIAS DE HAVENDER REQUIERE QUE CUANDO A UN PROCESO QUE MANTIENE RECURSOS LE ES NEGADA UNA PETICION DE RECURSOS ADICIONALES:
 - u DEBERA LIBERAR SUS RECURSOS Y SI ES NECESARIO PEDIRLOS DE NUEVO JUNTO CON LOS RECURSOS ADICIONALES.
- v LA IMPLEMENTACION DE ESTA ESTRATEGIA NIEGA LA CONDICION DE “NO APROPIACION”:
 - u LOS RECURSOS PUEDEN SER RETIRADOS DE LOS PROCESOS QUE LOS RETIENEN ANTES DE LA TERMINACION DE LOS PROCESOS.
- v EL PROBLEMA CONSISTE EN QUE EL RETIRO DE CIERTOS RECURSOS DE UN PROCESO PUEDE SIGNIFICAR:
 - u LA PERDIDA DEL TRABAJO EFECTUADO HASTA ESE PUNTO.
 - u LA NECESIDAD DE REPETIRLO LUEGO.
- v UNA CONSECUENCIA SERIA ES LA POSIBLE POSTERGACION INDEFINIDA DE UN PROCESO.

PREVENCION DE BLOQUEOS

- v **PREVENCION DE LA CONDICION DE “ESPERA CIRCULAR”.**
- v UNA FORMA ES QUE UN PROCESO SOLO ESTA AUTORIZADO A UTILIZAR UN RECURSO EN CADA MOMENTO:
 - u SI NECESITA OTRO RECURSOS, DEBE LIBERAR EL PRIMERO:
 - u ESTO RESULTA INACEPTABLE PARA MUCHOS PROCESOS.
- v OTRA FORMA ES LA SIGUIENTE:
 - u TODOS LOS RECURSOS SE NUMERAN GLOBALMENTE.
 - u LOS PROCESOS PUEDEN SOLICITAR LOS RECURSOS EN CUALQUIER MOMENTO:
 - Φ LAS SOLICITUDES SE DEBEN HACER SEGUN UN CIERTO ORDEN NUMERICO (CRECIENTE) DE RECURSO:
 - LA GRAFICA DE ASIGNACION DE RECURSOS NO TENDRA CICLOS.
 - u EN CADA INSTANTE UNO DE LOS RECURSOS ASIGNADOS TENDRA EL N° MAS GRANDE:
 - Φ EL PROCESO QUE LO POSEA NO PEDIRA UN RECURSO YA ASIGNADO.
 - Φ EL PROCESO TERMINARA O SOLICITARA RECURSOS CON N° MAYORES , QUE ESTARAN DISPONIBLES:
 - AL CONCLUIR LIBERARA SUS RECURSOS.
 - OTRO PROCESO TENDRA EL RECURSO CON EL N° MAYOR Y TAMBIEN PODRA TERMINAR.
 - TODOS LOS PROCESOS PODRAN TERMINAR Y NO HABRA BLOQUEO.
- v UNA VARIANTE CONSISTE EN ELIMINAR EL REQUISITO DE ADQUISICION DE RECURSOS EN ORDEN CRECIENTE:
 - u NINGUN PROCESO DEBE SOLICITAR UN RECURSO CON N° MENOR AL QUE POSEE EN EL MOMENTO.
- v EL PROBLEMA ES QUE EN CASOS REALES PODRIA RESULTAR IMPOSIBLE ENCONTRAR UN ORDEN QUE SATISFAGA A TODOS LOS PROCESOS.

- v LOS METODOS PARA PREVENIR EL BLOQUEO PUEDEN RESUMIRSE COMO SIGUE:
 - u CONDICION: EXCLUSION MUTUA.
 - u METODO: REALIZAR UN SPOOLING GENERAL.
 - u CONDICION: DETENERSE Y ESPERAR.
 - u METODO: SOLICITAR TODOS LOS RECURSOS AL PRINCIPIO.
 - u CONDICION: NO APROPIACION.
 - u METODO: RETIRAR LOS RECURSOS.
 - u CONDICION: ESPERA CIRCULAR.
 - u METODO: ORDENAR LOS RECURSOS EN FORMA NUMERICA.
- v OTROS ASPECTOS INTERESANTES RELACIONADOS CON BLOQUEOS SON:
 - u LA CERRADURA DE DOS FASES.
 - u LOS BLOQUEOS SIN RECURSOS.
 - u LA INANICION.
- v **CERRADURA DE DOS FASES.**
- v UNA OPERACION FRECUENTE EN SISTEMAS DE BASES DE DATOS CONSISTE EN:
 - u SOLICITAR EL CIERRE DE VARIOS REGISTROS.
 - u ACTUALIZAR TODOS LOS REGISTROS CERRADOS.
 - u ANTE LA EJECUCION DE VARIOS PROCESOS AL MISMO TIEMPO, EXISTE UN GRAVE RIESGO DE BLOQUEO.
- v EL METODO DE LA CERRADURA DE DOS FASES CONSISTE EN:
 - u PRIMER FASE: EL PROCESO INTENTA CERRAR TODOS LOS REGISTROS NECESARIOS, UNO A LA VEZ.
 - u SEGUNDA FASE: SE ACTUALIZA Y SE LIBERAN LAS CERRADURAS.
 - u SI DURANTE LA PRIMER FASE SE NECESITA ALGUN REGISTRO YA CERRADO:
 - Φ EL PROCESO LIBERA TODAS LAS CERRADURAS Y COMIENZA EN LA PRIMER FASE NUEVAMENTE.
 - Φ GENERALMENTE ESTO NO RESULTA APLICABLE EN LA REALIDAD:
 - NO RESULTA ACEPTABLE DEJAR UN PROCESO A LA MITAD Y VOLVER A COMENZAR.
 - EL PROCESO PODRIA HABER ACTUALIZADO ARCHIVOS, ENVIADO MENSAJES EN LA RED, ETC.

- v **BLOQUEOS SIN RECURSOS.**
- v LOS BLOQUEOS TAMBIEN PUEDEN APARECER EN SITUACIONES QUE NO ESTAN RELACIONADAS CON LOS RECURSOS.
- v PUEDE OCURRIR QUE DOS PROCESOS SE BLOQUEEN EN ESPERA DE QUE EL OTRO REALICE CIERTA ACCION:
 - u EJ.: OPERACIONES EFECTUADAS SOBRE SEMAFOROS (INDICADORES O VARIABLES DE CONTROL) EN ORDEN INCORRECTO.
- v **INANICION.**
- v EN UN SISTEMA DINAMICO PERMANENTEMENTE HAY SOLICITUDES DE RECURSOS.
- v SE NECESITA UN CRITERIO (POLITICA) PARA DECIDIR:
 - u QUIEN OBTIENE CUAL RECURSO.
 - u EN QUE MOMENTO.
- v PODRIA SUCEDER QUE CIERTOS PROCESOS NUNCA LOGRARAN EL SERVICIO, AUN SIN ESTAR BLOQUEADOS:
 - u SE PRIVILEGIA EN EL USO DEL RECURSO A OTROS PROCESOS.
- v LA INANICION SE PUEDE EVITAR MEDIANTE EL CRITERIO DE ASIGNACION DE RECURSOS “EL PRIMERO EN LLEGAR ES EL PRIMERO EN DESPACHAR (SER ATENDIDO)”.
- v EL PROCESO QUE HA ESPERADO EL MAXIMO TIEMPO SE DESPACHARA A CONTINUACION:
 - u EN EL TRANSCURSO DEL TIEMPO, CUALQUIERA DE LOS PROCESOS DADOS:
 - Φ SERA EL MAS ANTIGUO.
 - Φ OBTENDRA EL RECURSO NECESARIO.

TENDENCIAS DEL TRATAMIENTO DEL BLOQUEO

- v GENERALMENTE LOS S. O. HAN CONSIDERADO AL BLOQUEO COMO UNA INCOMODIDAD LIMITADA.
- v MUCHOS S. O. IMPLEMENTAN METODOS BASICOS DE PREVENCION DE BLOQUEOS SUGERIDOS POR HAVENDER:
 - u LOS RESULTADOS SON SATISFACTORIOS EN GRAN N° DE CASOS.
- v LA TENDENCIA ES A QUE EL BLOQUEO TENGA UNA CONSIDERACION MUCHO MAYOR EN LOS NUEVOS S. O., DEBIDO A:
 - u ORIENTACION HACIA LA OPERACION ASINCRONICA EN PARALELO:
 - Φ INCREMENTO DEL MULTIPROCESAMIENTO Y DE LAS OPERACIONES CONCURRENTES.
 - u ASIGNACION DINAMICA DE RECURSOS:
 - Φ CAPACIDAD DE LOS PROCESOS DE ADQUIRIR Y LIBERAR RECURSOS SEGUN LAS NECESIDADES.
 - Φ IGNORANCIA A PRIORI DE LOS PROCESOS RESPECTO DE SUS NECESIDADES DE RECURSOS.
 - u CONSIDERACION DE LOS DATOS COMO UN RECURSO:
 - Φ SIGNIFICA INCREMENTAR LA CAPACIDAD DEL S. O. PARA ADMINISTRAR GRAN N° DE RECURSOS.

INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS

- v INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS
- v VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LOS CENTRALIZADOS
- v VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LAS PC INDEPENDIENTES
- v DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS
- v CONCEPTOS DE HARDWARE
- v MULTIPROCESADORES CON BASE EN BUSES
- v MULTIPROCESADORES CON CONMUTADOR
- v MULTICOMPUTADORAS CON BASE EN BUSES
- v MULTICOMPUTADORAS CON CONMUTADOR
- v CONCEPTOS DE SOFTWARE
- v SISTEMAS OPERATIVOS DE REDES
- v SISTEMAS REALMENTE DISTRIBUIDOS
- v SISTEMAS DE MULTIPROCESADOR CON TIEMPO COMPARTIDO
- v ASPECTOS DEL DISEÑO
- v TRANSPARENCIA
- v FLEXIBILIDAD
- v CONFIABILIDAD
- v DESEMPEÑO
- v ESCALABILIDAD

INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS

- v DESDE EL INICIO DE LA ERA DE LA COMPUTADORA MODERNA (1945), HASTA CERCA DE 1985, SOLO SE CONOCIA LA COMPUTACION CENTRALIZADA.
- v A PARTIR DE LA MITAD DE LA DECADA DE LOS OCHENTAS APARECEN DOS AVANCES TECNOLOGICOS FUNDAMENTALES:
 - v DESARROLLO DE MICROPROCESADORES PODEROSOS Y ECONOMICOS:
 - ❖ ARQUITECTURAS DE 8, 16, 32 Y 64 BITS.
 - v DESARROLLO DE REDES DE AREA LOCAL (LAN) DE ALTA VELOCIDAD:
 - ❖ POSIBILIDAD DE CONECTAR CIENTOS DE MAQUINAS A VELOCIDADES DE TRANSFERENCIA DE MILLONES DE BITS POR SEGUNDO (Mb/SEG).
- v APARECEN LOS **SISTEMAS DISTRIBUIDOS**, EN CONTRASTE CON LOS **SISTEMAS CENTRALIZADOS**.
- v LOS SISTEMAS DISTRIBUIDOS NECESITAN UN SOFTWARE DISTINTO AL DE LOS SISTEMAS CENTRALIZADOS.
- v LOS S. O. PARA SISTEMAS DISTRIBUIDOS HAN TENIDO IMPORTANTES DESARROLLOS PERO TODAVIA EXISTE UN LARGO CAMINO POR RECORRER.
- v LOS USUARIOS PUEDEN ACCEDER A UNA GRAN VARIEDAD DE RECURSOS COMPUTACIONALES:
 - v DE HARDWARE Y DE SOFTWARE.
 - v DISTRIBUIDOS ENTRE UN GRAN N° DE SISTEMAS COMPUTACIONALES CONECTADOS.
- v UN IMPORTANTE ANTECEDENTE DE LAS REDES DE COMPUTADORAS LO CONSTITUYE ARPANET, INICIADA EN 1968 EN LOS EE. UU.

VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LOS CENTRALIZADOS

- v UNA RAZON PARA LA TENDENCIA HACIA LA DESCENTRALIZACION ES LA ECONOMIA.
- v HERB GROSCH FORMULO LA QUE SE LLAMARIA "LEY DE GROSCH":
 - υ EL PODER DE COMPUTO DE UNA CPU ES PROPORCIONAL AL CUADRADO DE SU PRECIO:
 - Φ SI SE PAGA EL DOBLE SE OBTIENE EL CUADRUPLA DEL DESEMPEÑO.
 - υ FUE APLICABLE EN LOS AÑOS SETENTAS Y OCHENTAS A LA TECNOLOGIA MAINFRAME.
 - υ NO ES APLICABLE A LA TECNOLOGIA DEL MICROPROCESADOR:
 - Φ LA SOLUCION MAS EFICAZ EN CUANTO A COSTO ES LIMITARSE A UN GRAN N° DE CPU BARATOS REUNIDOS EN UN MISMO SISTEMA.
- v LOS SISTEMAS DISTRIBUIDOS GENERALMENTE TIENEN EN POTENCIA UNA PROPORCION PRECIO / DESEMPEÑO MUCHO MEJOR QUE LA DE UN UNICO SISTEMA CENTRALIZADO.
- v ALGUNOS AUTORES DISTINGUEN ENTRE:
 - υ *SISTEMAS DISTRIBUIDOS*: ESTAN DISEÑADOS PARA QUE MUCHOS USUARIOS TRABAJEN EN FORMA CONJUNTA.
 - υ *SISTEMAS PARALELOS*: ESTAN DISEÑADOS PARA LOGRAR LA MAXIMA RAPIDEZ EN UN UNICO PROBLEMA.
- v EN GENERAL SE CONSIDERAN *SISTEMAS DISTRIBUIDOS*, EN SENTIDO AMPLIO, A LOS SISTEMAS EN QUE:
 - υ EXISTEN VARIAS CPU CONECTADAS ENTRE SI.
 - υ LAS DISTINTAS CPU TRABAJAN DE MANERA CONJUNTA.
- v CIERTAS APLICACIONES SON DISTRIBUIDAS EN FORMA INHERENTE:
 - υ EJ.: SISTEMA DE AUTOMATIZACION DE UNA FABRICA:
 - Φ CONTROLA LOS ROBOTS Y MAQUINAS EN LA LINEA DE MONTAJE.
 - Φ CADA ROBOT O MAQUINA ES CONTROLADO POR SU PROPIA COMPUTADORA.
 - Φ LAS DISTINTAS COMPUTADORAS ESTAN INTERCONECTADAS.

VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LOS CENTRALIZADOS

- v UNA VENTAJA POTENCIAL DE UN SISTEMA DISTRIBUIDO ES UNA MAYOR CONFIABILIDAD:
 - υ AL DISTRIBUIR LA CARGA DE TRABAJO EN MUCHAS MAQUINAS, LA FALLA DE UNA DE ELLAS NO AFECTARA A LAS DEMAS:
 - Φ LA CARGA DE TRABAJO PODRIA DISTRIBUIRSE.
 - υ SI UNA MAQUINA SE DESCOMPONE:
 - Φ SOBREVIVE EL SISTEMA COMO UN TODO.
- v OTRA VENTAJA IMPORTANTE ES LA POSIBILIDAD DEL CRECIMIENTO INCREMENTAL O POR INCREMENTOS:
 - υ PODRIAN AÑADIRSE PROCESADORES AL SISTEMA, PERMITIENDO UN DESARROLLO GRADUAL SEGUN LAS NECESIDADES.
 - υ NO SON NECESARIOS GRANDES INCREMENTOS DE POTENCIA EN BREVES LAPROS DE TIEMPO.
 - υ SE PUEDE AÑADIR PODER DE COMPUTO EN PEQUEÑOS INCREMENTOS.

VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LAS PC INDEPENDIENTES

- v SATISFACEN LA NECESIDAD DE MUCHOS USUARIOS DE COMPARTIR CIERTOS DATOS:
 - u EJ.: SISTEMA DE RESERVAS DE LINEAS AEREAS.
- v TAMBIEN CON LOS SISTEMAS DISTRIBUIDOS SE PUEDEN COMPARTIR OTROS RECURSOS COMO PROGRAMAS Y PERIFERICOS COSTOSOS:
 - u EJ.: IMPRESORAS LASER COLOR, EQUIPOS DE FOTOCOMPOSICION, DISPOSITIVOS DE ALMACENAMIENTO MASIVO (EJ.: CAJAS OPTICAS), ETC.
- v OTRA IMPORTANTE RAZON ES LOGRAR UNA MEJOR COMUNICACION ENTRE LAS PERSONAS:
 - u EJ.: CORREO ELECTRONICO:
 - Φ POSEE IMPORTANTES VENTAJAS SOBRE EL CORREO POR CARTAS, EL TELEFONO Y EL FAX:
 - VELOCIDAD, DISPONIBILIDAD, GENERACION DE DOCUMENTOS EDITABLES POR PROCESADORES DE TEXTO, ETC.
- v LA MAYOR FLEXIBILIDAD ES TAMBIEN IMPORTANTE:
 - u LA CARGA DE TRABAJO SE PUEDE DIFUNDIR (DISTRIBUIR) ENTRE LAS MAQUINAS DISPONIBLES EN LA FORMA MAS EFICAZ SEGUN EL CRITERIO ADOPTADO (POR EJ. COSTOS).
 - u LOS EQUIPOS DISTRIBUIDOS PUEDEN NO SER SIEMPRE PC:
 - Φ SE PUEDEN ESTRUCTURAR SISTEMAS CON GRUPOS DE PC Y DE COMPUTADORAS COMPARTIDAS, DE DISTINTA CAPACIDAD.

DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS

- v EL PRINCIPAL PROBLEMA ES EL SOFTWARE:
 - u EL DISEÑO, IMPLANTACION Y USO DEL SOFTWARE DISTRIBUIDO PRESENTA NUMEROSOS INCONVENIENTES.
- v LOS PRINCIPALES INTERROGANTES SON LOS SIGUIENTES:
 - u ¿QUE TIPO DE S. O., LENGUAJE DE PROGRAMACION Y APLICACIONES SON ADECUADOS PARA ESTOS SISTEMAS?.
 - u ¿CUANTO DEBEN SABER LOS USUARIOS DE LA DISTRIBUCION?.
 - u ¿QUE TANTO DEBE HACER EL SISTEMA Y QUE TANTO DEBEN HACER LOS USUARIOS?.
- v LA RESPUESTA A ESTOS INTERROGANTES NO ES UNIFORME ENTRE LOS ESPECIALISTAS:
 - u EXISTE UNA GRAN DIVERSIDAD DE CRITERIOS Y DE INTERPRETACIONES AL RESPECTO.
- v OTRO PROBLEMA POTENCIAL TIENE QUE VER CON LAS REDES DE COMUNICACIONES:
 - u SE DEBEN CONSIDERAR PROBLEMAS DEBIDOS A PERDIDAS DE MENSAJES, SATURACION EN EL TRAFICO, EXPANSION, ETC.
- v EL HECHO DE QUE SEA FACIL COMPARTIR LOS DATOS ES UNA VENTAJA PERO SE PUEDE CONVERTIR EN UN GRAN PROBLEMA:
 - u LA SEGURIDAD DEBE ORGANIZARSE ADECUADAMENTE.
- v EN GENERAL SE CONSIDERA QUE LAS VENTAJAS SUPERAN A LAS DESVENTAJAS, SI ESTAS ULTIMAS SE ADMINISTRAN SERIAMENTE.

CONCEPTOS DE HARDWARE

- v TODOS LOS SISTEMAS DISTRIBUIDOS CONSTAN DE VARIAS CPU, ORGANIZADAS DE DIVERSAS FORMAS, ESPECIALMENTE RESPECTO DE:
 - u LA FORMA DE INTERCONECTARLAS ENTRE SI.
 - u LOS ESQUEMAS DE COMUNICACION UTILIZADOS.
- v EXISTEN DIVERSOS ESQUEMAS DE CLASIFICACION PARA LOS SISTEMAS DE COMPUTOS CON VARIAS CPU:
 - u UNO DE LOS MAS CONOCIDOS ES LA **TAXONOMIA DE FLYNN**:
 - Φ CONSIDERA COMO CARACTERISTICAS ESENCIALES EL N° DE FLUJO DE INSTRUCCIONES Y EL N° DE FLUJOS DE DATOS.
 - Φ LA CLASIFICACION INCLUYE EQUIPOS **SISD**, **SIMD**, **MISD** Y **MIMD**.
- v SISD (SINGLE INSTRUCTION SINGLE DATA: UN FLUJO DE INSTRUCCIONES Y UN FLUJO DE DATOS):
 - u POSEEN UN UNICO PROCESADOR.
- v SIMD (SINGLE INSTRUCTION MULTIPLE DATA: UN FLUJO DE INSTRUCCIONES Y VARIOS FLUJOS DE DATOS):
 - u SE REFIERE A ORDENAR PROCESADORES CON UNA UNIDAD DE INSTRUCCION QUE:
 - Φ BUSCA UNA INSTRUCCION.
 - Φ INSTRUYE A VARIAS UNIDADES DE DATOS PARA QUE LA LLEVEN A CABO EN PARALELO, C / U CON SUS PROPIOS DATOS.
 - u SON UTILES PARA LOS COMPUTOS QUE REPITEN LOS MISMOS CALCULOS EN VARIOS CONJUNTOS DE DATOS.
- v MISD (MULTIPLE INSTRUCTION SINGLE DATA: UN FLUJO DE VARIAS INSTRUCCIONES Y UN SOLO FLUJO DE DATOS):
 - u NO SE PRESENTA EN LA PRACTICA.
- v MIMD (MULTIPLE INSTRUCTION MULTIPLE DATA: UN GRUPO DE COMPUTADORAS INDEPENDIENTES, C / U CON SU PROPIO CONTADOR DEL PROGRAMA, PROGRAMA Y DATOS):
 - u TODOS LOS SISTEMAS DISTRIBUIDOS SON DE ESTE TIPO.

CONCEPTOS DE HARDWARE

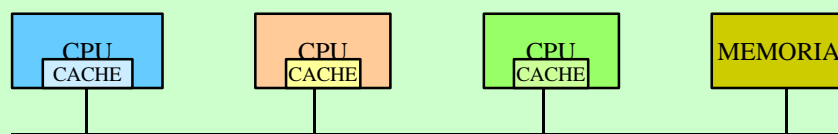
- v UN AVANCE SOBRE LA CLASIFICACION DE FLYNN INCLUYE LA DIVISION DE LAS COMPUTADORAS **MIMD** EN DOS GRUPOS:
 - v **MULTIPROCESADORES:** POSEEN MEMORIA COMPARTIDA:
 - ⊕ LOS DISTINTOS PROCESADORES COMPARTEN EL MISMO ESPACIO DE DIRECCIONES VIRTUALES.
 - v **MULTICOMPUTADORAS:** NO POSEEN MEMORIA COMPARTIDA:
 - ⊕ EJ.: GRUPO DE PC CONECTADAS MEDIANTE UNA RED.
- v C / U DE LAS CATEGORIAS INDICADAS SE PUEDE CLASIFICAR SEGUN LA ARQUITECTURA DE LA RED DE INTERCONEXION EN:
 - v **ESQUEMA DE BUS:**
 - ⊕ EXISTE UNA SOLA RED, BUS, CABLE U OTRO MEDIO QUE CONECTA TODAS LAS MAQUINAS:
 - EJ.: LA TELEVISION POR CABLE.
 - v **ESQUEMA CON CONMUTADOR:**
 - ⊕ NO EXISTE UNA SOLA COLUMNA VERTEBRAL DE CONEXION:
 - HAY MULTIPLES CONEXIONES Y VARIOS PATRONES DE CONEXIONADO.
 - LOS MENSAJES DE MUEVEN A TRAVES DE LOS MEDIOS DE CONEXION.
 - SE DECIDE EXPLICITAMENTE LA CONMUTACION EN CADA ETAPA PARA DIRIGIR EL MENSAJE A LO LARGO DE UNO DE LOS CABLES DE SALIDA.
 - EJ.: EL SISTEMA MUNDIAL TELEFONICO PUBLICO.
- v OTRO ASPECTO DE LA CLASIFICACION CONSIDERA EL ACOPLAMIENTO ENTRE LOS EQUIPOS:
 - v **SISTEMAS FUERTEMENTE ACOPLADOS:**
 - ⊕ EL RETRASO AL ENVIAR UN MENSAJE DE UNA COMPUTADORA A OTRA ES CORTO Y LA TASA DE TRANSMISION ES ALTA.
 - ⊕ GENERALMENTE SE LOS UTILIZA COMO SISTEMAS PARALELOS.
 - v **SISTEMAS DEBILMENTE ACOPLADOS:**
 - ⊕ EL RETRASO DE LOS MENSAJES ENTRE LAS MAQUINAS ES GRANDE Y LA TASA DE TRANSMISION ES BAJA.
 - ⊕ GENERALMENTE SE LOS UTILIZA COMO SISTEMAS DISTRIBUIDOS.
- v GENERALMENTE LOS MULTIPROCESADORES ESTAN MAS FUERTEMENTE ACOPLADOS QUE LAS MULTICOMPUTADORAS.

MULTIPROCESADORES CON BASE EN BUSES

- v CONSTAN DE CIERTO N° DE CPU CONECTADAS A UN BUS COMUN, JUNTO CON UN MODULO DE MEMORIA.
- v UN BUS TIPICO POSEE AL MENOS:
 - u 32 LINEAS DE DIRECCIONES.
 - u 32 LINEAS DE DATOS.
 - u 30 LINEAS DE CONTROL.
- v TODOS LOS ELEMENTOS PRECEDENTES OPERAN EN PARALELO.
- v PARA LEER UNA PALABRA DE MEMORIA, UNA CPU:
 - u COLOCA LA DIRECCION DE LA PALABRA DESEADA EN LAS LINEAS DE DIRECCIONES DEL BUS.
 - u COLOCA UNA SEÑAL EN LAS LINEAS DE CONTROL ADECUADAS PARA INDICAR QUE DESEA LEER.
 - u LA MEMORIA RESPONDE Y COLOCA EL VALOR DE LA PALABRA EN LAS LINEAS DE DATOS PARA PERMITIR LA LECTURA DE ESTA POR PARTE DE LA CPU SOLICITANTE.
- v PARA GRABAR EL PROCEDIMIENTO ES SIMILAR.
- v SOLO EXISTE UNA MEMORIA, LA CUAL PRESENTA LA PROPIEDAD DE LA **COHERENCIA**:
 - u LAS MODIFICACIONES HECHAS POR UNA CPU SE REFLEJAN DE INMEDIATO EN LAS SUBSIGUIENTES LECTURAS DE LA MISMA O DE OTRA CPU.
- v EL PROBLEMA DE ESTE ESQUEMA ES QUE EL BUS TIENDE A SOBRECARGARSE Y EL RENDIMIENTO A DISMINUIR DRASTICAMENTE:
 - u LA SOLUCION ES AÑADIR UNA **MEMORIA CACHE** DE ALTA VELOCIDAD ENTRE LA CPU Y EL BUS:
 - Φ EL CACHE GUARDA LAS PALABRAS DE ACCESO RECIENTE.
 - Φ TODAS LAS SOLICITUDES DE LA MEMORIA PASAN A TRAVES DEL CACHE.
 - Φ SI LA PALABRA SOLICITADA SE ENCUENTRA EN EL CACHE:
 - EL CACHE RESPONDE A LA CPU.
 - NO SE HACE SOLICITUD ALGUNA AL BUS.
 - u SI EL CACHE ES LO BASTANTE GRANDE:
 - Φ LA **TASA DE ENCUENTROS** SERA ALTA Y LA CANTIDAD DE TRAFICO EN EL BUS POR CADA CPU DISMINUIRA DRASTICAMENTE:
 - PERMITE INCREMENTAR EL N° DE CPU.

MULTIPROCESADORES CON BASE EN BUSES

MULTIPROCESADORES CON BASE EN UN BUS

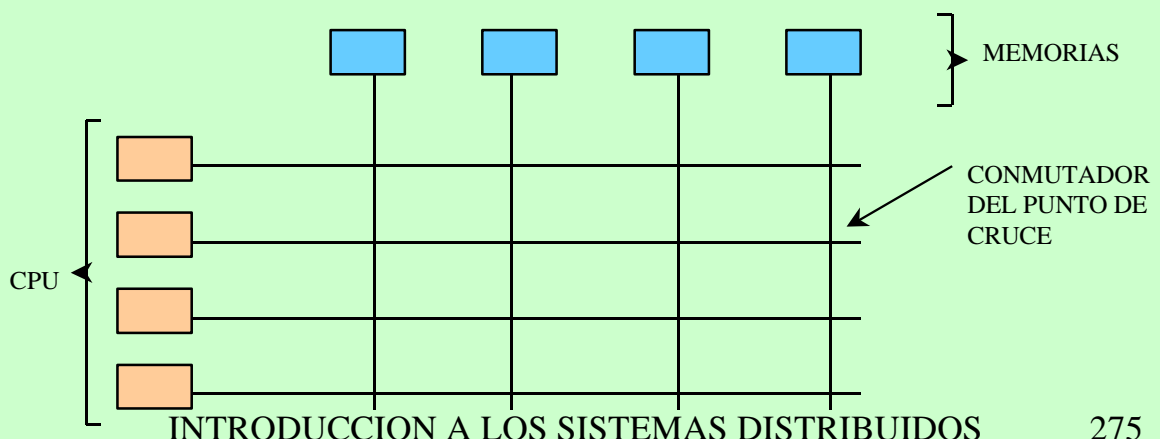


- v UN IMPORTANTE PROBLEMA DEBIDO AL USO DE CACHES ES EL DE LA **INCOHERENCIA DE LA MEMORIA**:
 - SUPONGAMOS QUE LAS CPU A Y B LEEN LA MISMA PALABRA DE MEMORIA EN SUS RESPECTIVOS CACHES.
 - A ESCRIBE SOBRE LA PALABRA.
 - CUANDO B LEE ESA PALABRA, OBTIENE UN VALOR ANTERIOR Y NO EL VALOR RECIEN ACTUALIZADO POR A.
- v UNA SOLUCION CONSISTE EN LO SIGUIENTE:
 - DISEÑAR LAS CACHE DE TAL FORMA QUE *CUANDO UNA PALABRA SEA ESCRITA AL CACHE, TAMBIEN SEA ESCRITA A LA MEMORIA*.
 - A ESTO SE DENOMINA **CACHE DE ESCRITURA**.
 - NO CAUSA TRAFICO EN EL BUS EL USO DE CACHE PARA LA LECTURA.
 - SI CAUSA TRAFICO EN EL BUS:
 - ⊕ EL NO USO DE CACHE PARA LA LECTURA.
 - ⊕ TODA LA ESCRITURA.
- v SI TODOS LOS CACHES REALIZAN UN MONITOREO CONSTANTE DEL BUS:
 - CADA VEZ QUE UN CACHE OBSERVA UNA ESCRITURA A UNA DIRECCION DE MEMORIA PRESENTE EN EL:
 - ⊕ PUEDE ELIMINAR ESE DATO O ACTUALIZARLO EN EL CACHE CON EL NUEVO VALOR.
 - ESTOS CACHES SE DENOMINAN **CACHES MONITORES**.
- v *UN DISEÑO CON CACHES MONITORES Y DE ESCRITURA ES COHERENTE E INVISIBLE PARA EL PROGRAMADOR*:
 - ES MUY UTILIZADO EN MULTIPROCESADORES BASADOS EN BUSES.

MULTIPROCESADORES CON CONMUTADOR

- v EL ESQUEMA DE MULTIPROCESADORES CON BASE EN BUSES RESULTA APROPIADO PARA HASTA APROXIMADAMENTE 64 PROCESADORES.
- v PARA SUPERAR ESTA CIFRA ES NECESARIO UN METODO DISTINTO DE CONEXION ENTRE PROCESADORES (CPU) Y MEMORIA.
- v UNA POSIBILIDAD ES DIVIDIR LA MEMORIA EN MODULOS Y CONECTARLOS A LAS CPU CON UN **CONMUTADOR DE CRUCETA (CROSS-BAR SWITCH)**:
 - v CADA CPU Y CADA MEMORIA TIENE UNA CONEXION QUE SALE DE EL.
 - v EN CADA INTERSECCION ESTA UN **CONMUTADOR DEL PUNTO DE CRUCE (CROSSPOINT SWITCH)** ELECTRONICO QUE EL HARDWARE PUEDE ABRIR Y CERRAR:
 - Φ CUANDO UNA CPU DESEA TENER ACCESO A UNA MEMORIA PARTICULAR, EL CONMUTADOR DEL PUNTO DE CRUCE QUE LOS CONECTA SE CIERRA MOMENTANEAMENTE.
 - v LA VIRTUD DEL CONMUTADOR DE CRUCETA ES QUE MUCHAS CPU PUEDEN TENER ACCESO A LA MEMORIA AL MISMO TIEMPO:
 - Φ AUNQUE NO A LA MISMA MEMORIA SIMULTANEAMENTE.
 - v LO NEGATIVO DE ESTE ESQUEMA ES EL ALTO N° DE CONMUTADORES:
 - Φ PARA n CPU Y n MEMORIAS SE NECESITAN $n \times n$ CONMUTADORES.

CONMUTADOR DE CRUCETA

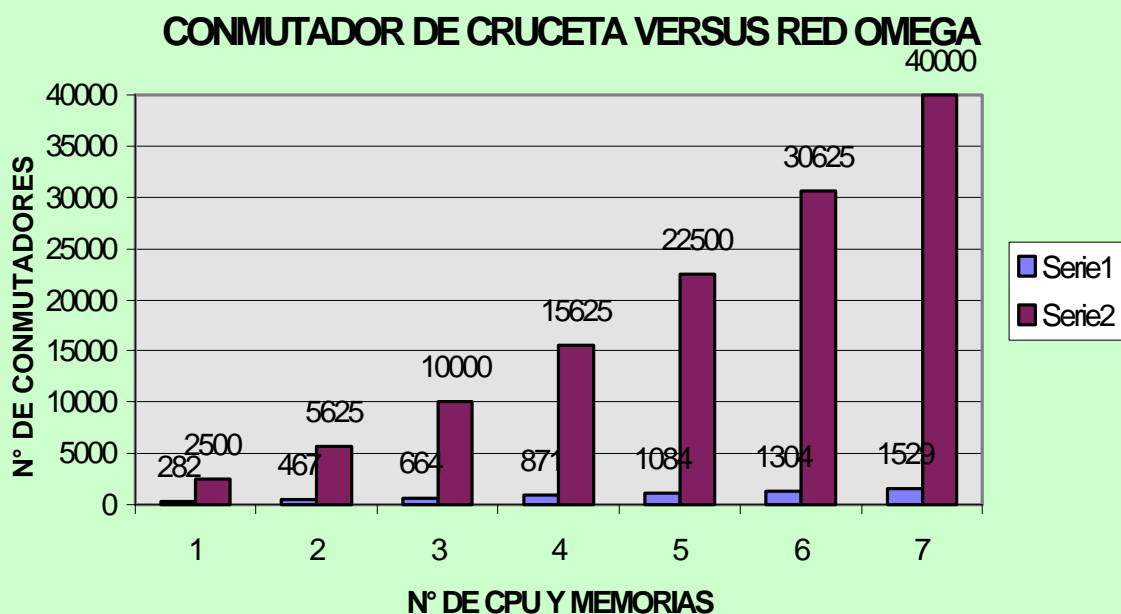


MULTIPROCESADORES CON CONMUTADOR

- v EL N° DE CONMUTADORES DEL ESQUEMA ANTERIOR PUEDE RESULTAR PROHIBITIVO:
 - u OTROS ESQUEMAS PRECISAN MENOS CONMUTADORES, POR EJ., LA **RED OMEGA**:
 - Φ POSEE CONMUTADORES 2 X 2:
 - C / U TIENE 2 ENTRADAS Y 2 SALIDAS.
 - CADA CONMUTADOR PUEDE DIRIGIR CUALQUIERA DE LAS ENTRADAS EN CUALQUIERA DE LAS SALIDAS.
 - ELIGIENDO LOS ESTADOS ADECUADOS DE LOS CONMUTADORES:
 - CADA CPU PODRA TENER ACCESO A CADA MEMORIA.
 - Φ PARA n CPU Y n MEMORIAS SE PRECISAN:
 - n ETAPAS DE CONMUTACION.
 - CADA ETAPA TIENE $\log_2 n$ CONMUTADORES PARA UN TOTAL DE $n \log_2 n$ CONMUTADORES:
 - ESTE N° ES MENOR QUE $n \times n$ DEL ESQUEMA ANTERIOR, PERO SIGUE SIENDO MUY GRANDE PARA n GRANDE.

n	LOG (n,2)	n * LOG (n,2)	n * n
50	5,64385619	282	2500
75	6,22881869	467	5625
100	6,64385619	664	10000
125	6,96578428	871	15625
150	7,22881869	1084	22500
175	7,45121111	1304	30625
200	7,64385619	1529	40000
1024	10	10240	1048576

MULTIPROCESADORES CON CONMUTADOR



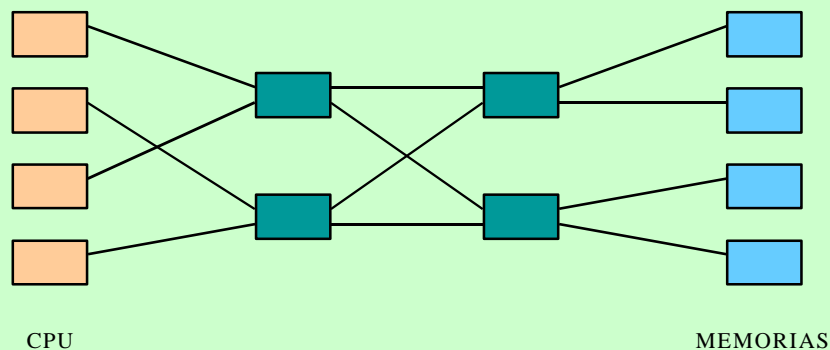
ACLARACIONES:

- * SERIE 1: CORRESPONDE A RED OMEGA.
- * SERIE 2: CORRESPONDE A CONMUTADOR DE CRUCETA.
- * LOS VALORES SEÑALADOS EN N° DE CPU Y MEMORIAS CORRESPONDEN A LA SIGUIENTE EQUIVALENCIA:

1:	50
2:	75
3:	100
4:	125
5:	150
6:	175
7:	200

MULTIPROCESADORES CON CONMUTADOR

RED OMEGA DE CONMUTACION

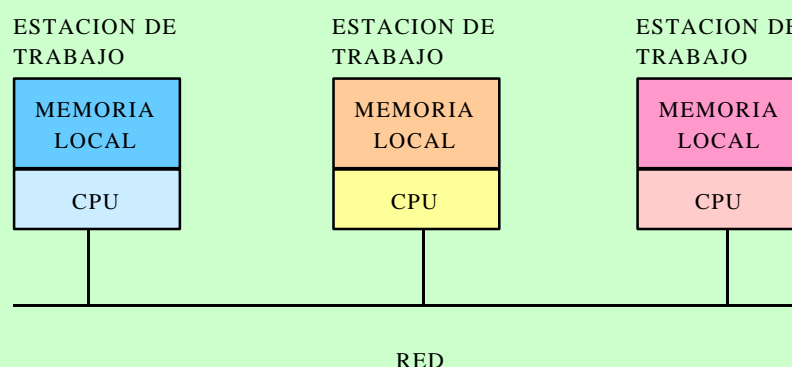


- v UN PROBLEMA IMPORTANTE EN LA RED OMEGA ES EL RETRASO:
 - v EJ.: SI $n = 1024$ EXISTEN SEGUN LA TABLA ANTERIOR:
 - ⊕ 10 ETAPAS DE CONMUTACION DE LA CPU A LA MEMORIA.
 - ⊕ 10 ETAPAS PARA QUE LA PALABRA SOLICITADA DE LA MEMORIA REGRESE.
 - ⊕ SI LA CPU ES DE 50 MHZ, EL TIEMPO DE EJECUCION DE UNA INSTRUCCION ES DE 20 NSEG.
 - ⊕ SI UNA SOLICITUD DE LA MEMORIA DEBE RECORRER 20 ETAPAS DE CONMUTACION (10 DE IDA Y 10 DE REGRESO) EN 20 NSEG:
 - EL TIEMPO DE CONMUTACION DEBE SER DE 1 NSEG.
 - EL MULTIPROCESADOR DE 1024 CPU NECESITARA 10240 CONMUTADORES DE 1 NSEG.
 - EL COSTO SERA ALTO.
- v OTRA POSIBLE SOLUCION SON LOS ESQUEMAS SEGUN SISTEMAS JERARQUICOS:
 - v CADA CPU TIENE ASOCIADA CIERTA MEMORIA LOCAL.
 - v EL ACCESO SERA MUY RAPIDO A LA PROPIA MEMORIA LOCAL Y MAS LENTO A LA MEMORIA DE LAS DEMAS CPU.
 - v ESTO SE DENOMINA ESQUEMA O MAQUINA NUMA (ACCESO NO UNIFORME A LA MEMORIA):
 - ⊕ TIENEN UN MEJOR TIEMPO PROMEDIO DE ACCESO QUE LAS MAQUINAS BASADAS EN REDES OMEGA.
 - ⊕ LA COLOCACION DE LOS PROGRAMAS Y DATOS EN MEMORIA ES CRITICA PARA LOGRAR QUE LA MAYORIA DE LOS ACCESOS SEAN A LA MEMORIA LOCAL DE CADA CPU.

MULTICOMPUTADORAS CON BASE EN BUSES

- v ES UN ESQUEMA SIN MEMORIA COMPARTIDA.
- v CADA CPU TIENE UNA CONEXION DIRECTA CON SU PROPIA MEMORIA LOCAL.
- v UN PROBLEMA IMPORTANTE ES LA FORMA EN QUE LAS CPU SE COMUNIQUEN ENTRE SI.
- v EL TRAFICO ES SOLO ENTRE UNA CPU Y OTRA:
 - u EL VOLUMEN DE TRAFICO SERA VARIOS ORDENES DE MAGNITUD MENOR QUE SI SE UTILIZARA LA RED DE INTERCONEXION PARA EL TRAFICO CPU - MEMORIA.
- v TOPOLOGICAMENTE ES UN ESQUEMA SIMILAR AL DEL MULTIPROCESADOR BASADO EN UN BUS.
- v CONSISTE GENERALMENTE EN UNA COLECCION DE ESTACIONES DE TRABAJO EN UNA LAN (RED DE AREA LOCAL).

EJ. DE MULTICOMPUTADORA QUE CONSTA DE ESTACIONES DE TRABAJO EN UNA LAN

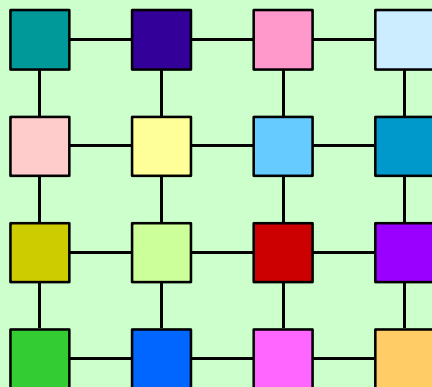


MULTICOMPUTADORAS CON CONMUTADOR

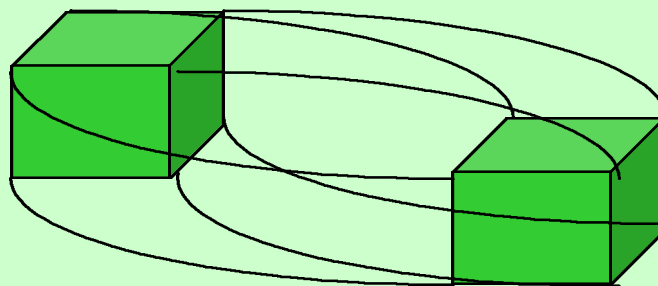
- v CADA CPU TIENE ACCESO DIRECTO Y EXCLUSIVO A SU PROPIA MEMORIA PARTICULAR.
- v EXISTEN DIVERSAS TOPOLOGIAS, LAS MAS COMUNES SON LA *RETICULA* Y EL *HIPERCUBO*.
- v LAS PRINCIPALES CARACTERISTICAS DE LAS RETICULAS SON:
 - u SON FACILES DE COMPRENDER.
 - u SE BASAN EN LAS TARJETAS DE CIRCUITOS IMPRESOS.
 - u SE ADECUAN A PROBLEMAS CON UNA NATURALEZA BIDIMENSIONAL INHERENTE (TEORIA DE GRAFICAS, VISION ARTIFICIAL, ETC.).
- v LAS PRINCIPALES CARACTERISTICAS DEL HIPERCUBO SON:
 - u ES UN CUBO n - DIMENSIONAL.
 - u EN UN HIPERCUBO DE DIMENSION 4:
 - Φ SE PUEDE CONSIDERAR COMO DOS CUBOS ORDINARIOS, C / U DE ELLOS CON 8 VERTICES Y 12 ARISTAS.
 - Φ CADA VERTICE ES UN CUBO.
 - Φ CADA ARISTA ES UNA CONEXION ENTRE 2 CPU.
 - Φ SE CONECTAN LOS VERTICES CORRESPONDIENTES DE C / U DE LOS CUBOS.
 - u EN UN HIPERCUBO DE DIMENSION 5:
 - Φ SE DEBERIAN AÑADIR DOS CUBOS CONECTADOS ENTRE SI Y CONECTAR LAS ARISTAS CORRESPONDIENTES EN LAS DOS MITADES, Y ASI SUCESIVAMENTE.
 - u EN UN HIPERCUBO DE n DIMENSIONES:
 - Φ CADA CPU TIENE n CONEXIONES CON OTRAS CPU.
 - Φ LA COMPLEJIDAD DEL CABLEADO AUMENTA EN PROPORCION LOGARITMICA CON EL TAMAÑO.
 - Φ SOLO SE CONECTAN LOS PROCESADORES VECINOS MAS CERCANOS:
 - MUCHOS MENSAJES DEBEN REALIZAR VARIOS SALTOS ANTES DE LLEGAR A SU DESTINO.
 - LA TRAYECTORIA MAS GRANDE CRECE EN FORMA LOGARITMICA CON EL TAMAÑO:
 - EN LA RETICULA CRECE COMO LA RAIZ CUADRADA DEL N° DE CPU.
 - Φ CON LA TECNOLOGIA ACTUAL YA SE PUEDEN PRODUCIR HIPERCUBOS DE 16384 CPU.

MULTICOMPUTADORAS CON CONMUTADOR

RETICULA



HIPERCUBO DE DIMENSION 4



CONCEPTOS DE SOFTWARE

- v LA IMPORTANCIA DEL SOFTWARE SUPERA FRECUENTEMENTE A LA DEL HARDWARE.
- v LA IMAGEN QUE UN SISTEMA PRESENTA QUEDA DETERMINADA EN GRAN MEDIDA POR EL SOFTWARE DEL S. O. Y NO POR EL HARDWARE.
- v LOS S. O. NO SE PUEDEN ENCASILLAR FACILMENTE, COMO EL HARDWARE, PERO SE LOS PUEDE CLASIFICAR EN DOS TIPOS:
 - v DEBILMENTE ACOPLADOS.
 - v FUERTEMENTE ACOPLADOS.
- v EL SOFTWARE DEBILMENTE ACOPLADO DE UN SISTEMA DISTRIBUIDO:
 - v PERMITE QUE LAS MAQUINAS Y USUARIOS SEAN INDEPENDIENTES ENTRE SI EN LO FUNDAMENTAL.
 - v FACILITA QUE INTERACTUEN EN CIERTO GRADO CUANDO SEA NECESARIO.
 - v LOS EQUIPOS INDIVIDUALES SE DISTINGUEN FACILMENTE.
- v COMBINANDO LOS DISTINTOS TIPOS DE HARDWARE DISTRIBUIDO CON SOFTWARE DISTRIBUIDO SE LOGRAN DISTINTAS SOLUCIONES:
 - v NO TODAS INTERESAN DESDE EL PUNTO DE VISTA FUNCIONAL DEL USUARIO:
 - ❖ EJ.: UN MULTIPROCESADOR ES UN MULTIPROCESADOR:
 - NO IMPORTA SI UTILIZA UN BUS CON CACHES MONITORES O UNA RED OMEGA.

SISTEMAS OPERATIVOS DE REDES

- v UNA POSIBILIDAD ES EL SOFTWARE DEBILMENTE ACOPLADO EN HARDWARE DEBILMENTE ACOPLADO:
 - υ ES UNA SOLUCION MUY UTILIZADA.
 - υ EJ.: UNA RED DE ESTACIONES DE TRABAJO CONECTADAS MEDIANTE UNA LAN.
- v CADA USUARIO TIENE UNA ESTACION DE TRABAJO PARA SU USO EXCLUSIVO:
 - υ TIENE SU PROPIO S. O.
 - υ LA MAYORIA DE LOS REQUERIMIENTOS SE RESUELVEN LOCALMENTE.
 - υ ES POSIBLE QUE UN USUARIO SE CONECTE DE MANERA REMOTA CON OTRA ESTACION DE TRABAJO:
 - ⊕ MEDIANTE UN COMANDO DE “LOGIN REMOTO”.
 - ⊕ SE CONVIERTE LA PROPIA ESTACION DE TRABAJO DEL USUARIO EN UNA TERMINAL REMOTA ENLAZADA CON LA MAQUINA REMOTA.
 - ⊕ LOS COMANDOS SE ENVIAN A LA MAQUINA REMOTA.
 - ⊕ LA SALIDA DE LA MAQUINA REMOTA SE EXHIBE EN LA PANTALLA LOCAL.
 - υ PARA ALTERNAR CON OTRA MAQUINA REMOTA, PRIMERO HAY QUE DESCONECTARSE DE LA PRIMERA:
 - ⊕ EN CUALQUIER INSTANTE SOLO SE PUEDE UTILIZAR UNA MAQUINA.
 - υ LAS REDES TAMBIEN DISPONEN DE UN COMANDO DE COPIADO REMOTO DE ARCHIVOS DE UNA MAQUINA A OTRA:
 - ⊕ REQUIERE QUE EL USUARIO CONOZCA:
 - LA POSICION DE TODOS LOS ARCHIVOS.
 - EL SITIO DONDE SE EJECUTAN TODOS LOS COMANDOS.
- v UNA MEJOR SOLUCION CONSISTE EN UN *SISTEMA DE ARCHIVOS GLOBAL COMPARTIDO*, ACCESIBLE DESDE TODAS LAS ESTACIONES DE TRABAJO:
 - υ UNA O VARIAS MAQUINAS SOPORTAN AL SISTEMA DE ARCHIVOS:
 - ⊕ SON LOS **SERVIDORES DE ARCHIVOS**.

SISTEMAS OPERATIVOS DE REDES

v LOS **SERVIDORES DE ARCHIVOS**:

u ACEPTAN SOLICITUDES DE LOS PROGRAMAS DE USUARIOS:

- Φ LOS PROGRAMAS SE EJECUTAN EN LAS MAQUINAS NO SERVIDORAS, LLAMADAS **CLIENTES**.
- Φ LAS SOLICITUDES SE EXAMINAN, SE EJECUTAN Y LA RESPUESTA SE ENVIA DE REGRESO.

u GENERALMENTE TIENEN UN SISTEMA JERARQUICO DE ARCHIVOS.

v LAS ESTACIONES DE TRABAJO PUEDEN IMPORTAR O MONTAR ESTOS SISTEMAS DE ARCHIVOS:

u SE INCREMENTAN SUS SISTEMAS DE ARCHIVOS LOCALES.

u SE PUEDEN MONTAR LOS SERVIDORES EN LUGARES DIFERENTES DE SUS RESPECTIVOS SISTEMAS DE ARCHIVOS:

- Φ LAS RUTAS DE ACCESO A UN DETERMINADO ARCHIVO PUEDEN SER DIFERENTES PARA LAS DISTINTAS ESTACIONES.
- Φ LOS DISTINTOS CLIENTES TIENEN UN PUNTO DE VISTA DISTINTO DEL SISTEMA DE ARCHIVOS.
- Φ EL NOMBRE DE UN ARCHIVO DEPENDE:
 - DEL LUGAR DESDE EL CUAL SE TIENE ACCESO A EL.
 - DE LA CONFIGURACION DEL SISTEMA DE ARCHIVOS.

v EL S. O. DE ESTE TIPO DE AMBIENTE DEBE:

u CONTROLAR LAS ESTACIONES DE TRABAJO EN LO INDIVIDUAL.

u CONTROLAR A LOS SERVIDORES DE ARCHIVO.

u DEBE ENCARGARSE DE LA COMUNICACION ENTRE LOS SERVIDORES.

v TODAS LAS MAQUINAS PUEDEN EJECUTAR EL MISMO S. O., PERO ESTO NO ES NECESARIO.

v SI LOS CLIENTES Y LOS SERVIDORES EJECUTAN DIVERSOS S. O.:

u COMO MINIMO DEBEN COINCIDIR EN EL FORMATO Y SIGNIFICADO DE TODOS LOS MENSAJES QUE PODRIAN INTERCAMBIAR.

v ESQUEMAS COMO ESTE SE DENOMINAN **SISTEMA OPERATIVO DE RED**:

u CADA MAQUINA TIENE UN ALTO GRADO DE AUTONOMIA.

u EXISTEN POCOS REQUISITOS A LO LARGO DE TODO EL SISTEMA.

SISTEMAS OPERATIVOS DE REDES

- v **NFS: NETWORK FILE SYSTEM.**
- v ES UNO DE LOS MAS CONOCIDOS Y ACEPTADO COMO SISTEMA OPERATIVO DE RED.
- v FUE UN DESARROLLO DE SUN MICROSYSTEMS, SOPORTADO TAMBIEN POR DISTINTOS FABRICANTES:
 - v SURGIO PARA UNIX PERO SE AMPLIO A OTROS S. O. (EJ.: MS - DOS).
 - v SOPORTA SISTEMAS HETEROGENEOS:
 - ⊕ EJ.: CLIENTES DE MS - DOS QUE HAGAN USO DE SERVIDORES UNIX.
 - v LOS EQUIPOS PUEDEN SER TAMBIEN DE HARDWARE HETEROGENEO.
- v LOS ASPECTOS MAS INTERESANTES SON LOS RELACIONADOS CON:
 - v LA ARQUITECTURA.
 - v EL PROTOCOLO.
 - v LA IMPLANTACION.
- v *LA ARQUITECTURA DE NFS.*
- v LA IDEA FUNDAMENTAL ES PERMITIR QUE UNA COLECCION ARBITRARIA DE CLIENTES Y SERVIDORES COMPARTAN UN SISTEMA DE ARCHIVOS COMUN.
- v GENERALMENTE TODOS LOS CLIENTES Y SERVIDORES ESTAN EN LA MISMA LAN, PERO ESTO NO ES NECESARIO:
 - v SE PUEDE EJECUTAR NFS EN UNA RED DE AREA AMPLIA.
- v NFS PERMITE QUE CADA MAQUINA SEA UN CLIENTE Y UN SERVIDOR AL MISMO TIEMPO.
- v CADA SERVIDOR DE NFS EXPORTA UNO O VARIOS DE SUS DIRECTORIOS (Y SUBDIRECTORIOS DEPENDIENTES) PARA EL ACCESO POR PARTE DE CLIENTES REMOTOS.
- v LOS CLIENTES TIENEN ACCESO A LOS DIRECTORIOS EXPORTADOS MEDIANTE EL MONTAJE:
 - v CUANDO UN CLIENTE MONTA UN DIRECTORIO (REMOTO), ESTE SE CONVIERTE EN PARTE DE SU JERARQUIA DE DIRECTORIOS.

SISTEMAS OPERATIVOS DE REDES

- v UN CLIENTE SIN DISCO PUEDE MONTAR UN ARCHIVO REMOTO EN SU DIRECTORIO RAIZ:
 - u ESTO PRODUCE UN SISTEMA DE ARCHIVOS SOPORTADO EN SU TOTALIDAD EN UN SERVIDOR REMOTO.
- v LAS ESTACIONES DE TRABAJO QUE NO POSEEN DISCOS LOCALES PUEDEN MONTAR DIRECTORIOS REMOTOS EN DONDE LO DESEEN, EN LA PARTE SUPERIOR DE SU JERARQUIA DE DIRECTORIOS LOCAL:
 - u ESTA PRODUCE UN SISTEMA DE ARCHIVOS QUE ES EN PARTE LOCAL Y EN PARTE REMOTO.
- v SI DOS O MAS CLIENTES MONTAN EL MISMO DIRECTORIO AL MISMO TIEMPO:
 - u SE PUEDEN COMUNICAR AL COMPARTIR ARCHIVOS EN SUS DIRECTORIOS COMUNES.
 - u NO HAY QUE HACER NADA ESPECIAL PARA LOGRAR COMPARTIR LOS ARCHIVOS.
- v LOS ARCHIVOS COMPARTIDOS FIGURAN EN LA JERARQUIA DE DIRECTORIOS DE VARIAS MAQUINAS:
 - u SE LOS PUEDE LEER O ESCRIBIR DE LA MANERA USUAL.
- v *PROTOCOLOS DE NFS.*
- v UNO DE LOS OBJETIVOS DE NFS ES:
 - u SOPORTAR UN SISTEMA HETEROGENEO EN DONDE LOS CLIENTES Y SERVIDORES PODRIAN EJECUTAR DISTINTOS S. O. EN HARDWARE DIVERSO:
 - Φ ES ESENCIAL QUE LA INTERFAZ ENTRE LOS CLIENTES Y LOS SERVIDORES ESTE BIEN DEFINIDA.
- v NFS LOGRA ESTE OBJETIVO DEFINIENDO DOS PROTOCOLOS CLIENTE - SERVIDOR:
 - u UN **PROTOCOLO** ES UN CONJUNTO DE:
 - Φ SOLICITUDES QUE ENVIAN LOS CLIENTES A LOS SERVIDORES.
 - Φ LAS RESPUESTAS QUE ENVIAN LOS SERVIDORES DE REGRESO A LOS CLIENTES.

SISTEMAS OPERATIVOS DE REDES

- v UN PROTOCOLO DE NFS MANEJA EL **MONTAJE**.
- v UN CLIENTE PUEDE:
 - u ENVIAR EL NOMBRE DE UNA RUTA DE ACCESO A UN SERVIDOR.
 - u SOLICITAR EL PERMISO PARA MONTAR ESE DIRECTORIO EN ALGUNA PARTE DE SU JERARQUIA DE DIRECTORIOS.
- v SI EL NOMBRE DE LA RUTA DE ACCESO ES VALIDO Y EL DIRECTORIO ESPECIFICADO HA SIDO EXPORTADO:
 - u EL SERVIDOR REGRESA UN **ASA DE ARCHIVO (FILE HANDLE)** AL CLIENTE:
 - Φ CONTIENE CAMPOS QUE IDENTIFICAN:
 - DE MANERA UNICA EL TIPO DE SISTEMA DE ARCHIVOS, EL DISCO, EL N° DE NODO-I DEL DIRECTORIO.
 - LA INFORMACION RELATIVA A LA SEGURIDAD.
 - Φ ES UTILIZADA EN LLAMADAS POSTERIORES PARA LA LECTURA O ESCRITURA DE ARCHIVOS EN EL DIRECTORIO MONTADO.
- v ALGUNOS S. O. SOPORTAN LA ALTERNATIVA DEL **AUTOMONTAJE**:
 - u PERMITE QUE UN CONJUNTO DE DIRECTORIOS REMOTOS QUEDE ASOCIADO CON UN DIRECTORIO LOCAL.
 - u NINGUNO DE LOS DIRECTORIOS REMOTOS SE MONTA DURANTE EL ARRANQUE DEL CLIENTE.
 - u LA PRIMERA VEZ QUE SE ABRA UN ARCHIVO REMOTO, EL S. O. ENVIA UN MENSAJE A LOS SERVIDORES:
 - Φ LOS SERVIDORES RESPONDEN Y SE MONTA SU DIRECTORIO.
 - u LAS PRINCIPALES VENTAJAS SOBRE EL MONTAJE ESTATICO SON:
 - Φ SE EVITA EL TRABAJO DE CONTACTAR SERVIDORES Y MONTAR DIRECTORIOS QUE NO SON REQUERIDOS DE INMEDIATO.
 - Φ SI EL CLIENTE PUEDE UTILIZAR VARIOS SERVIDORES EN PARALELO, SE PUEDE TENER:
 - CIERTA TOLERANCIA A FALLAS.
 - MEJORAR EL RENDIMIENTO.

SISTEMAS OPERATIVOS DE REDES

- v NFS NO DA SOPORTE A LA DUPLICACION DE ARCHIVOS O DIRECTORIOS.
- v OTRO PROTOCOLO DE NFS ES PARA EL **ACCESO A LOS DIRECTORIOS Y ARCHIVOS**.
- v LOS CLIENTES PUEDEN:
 - u ENVIAR MENSAJES A LOS SERVIDORES PARA EL MANEJO DE LOS DIRECTORIOS Y LA LECTURA O ESCRITURA DE ARCHIVOS.
 - u TENER ACCESO A LOS ATRIBUTOS DE ARCHIVO, TALES COMO SU MODO, TAMAÑO Y FECHA DE LA ULTIMA MODIFICACION.
- v NFS SOPORTA **SERVIDORES SIN ESTADO**:
 - u NO MANTIENEN LA INFORMACION DE ESTADO RELATIVA A LOS ARCHIVOS ABIERTOS.
 - u SI UN SERVIDOR FALLA Y ARRANCA RAPIDAMENTE, NO SE PIERDE INFORMACION ACERCA DE LOS ARCHIVOS ABIERTOS:
 - Φ LOS PROGRAMAS CLIENTE NO FALLAN.
- v EL **SISTEMA DE ARCHIVOS REMOTOS (RFS)** DEL SISTEMA V DE UNIX NO FUNCIONA ASI, SINO QUE:
 - u EL SERVIDOR LLEVA UN REGISTRO DEL HECHO QUE CIERTO ARCHIVO ESTA ABIERTO Y LA POSICION ACTUAL DEL LECTOR.
 - u SI UN SERVIDOR FALLA Y VUELVE A ARRANCAR RAPIDAMENTE:
 - Φ SE PIERDEN TODAS LAS CONEXIONES ABIERTAS.
 - Φ LOS PROGRAMAS CLIENTE FALLAN.
- v EN UN SERVIDOR SIN ESTADO, COMO NFS:
 - u LOS BLOQUEOS NO TIENEN QUE ASOCIARSE CON LOS ARCHIVOS ABIERTOS:
 - Φ EL SERVIDOR NO SABE CUALES ARCHIVOS ESTAN ABIERTOS.
 - u SE NECESITA UN MECANISMO ADICIONAL INDEPENDIENTE PARA CONTROLAR EL BLOQUEO.

SISTEMAS OPERATIVOS DE REDES

- v NFS UTILIZA EL ESQUEMA DE PROTECCION DE UNIX, CON LOS BITS *rw**x* PARA EL PROPIETARIO, GRUPO Y OTROS.
- v SE PUEDE UTILIZAR LA CRIPTOGRAFIA DE CLAVES PUBLICAS PARA DAR VALIDEZ AL CLIENTE Y EL SERVIDOR EN CADA SOLICITUD Y RESPUESTA:
 - v EL CLIENTE MALICIOSO NO PUEDE PERSONIFICAR A OTRO CLIENTE, YA QUE NO CONOCE SU CLAVE SECRETA.
- v LAS CLAVES UTILIZADAS PARA LA AUTENTIFICACION, ASI COMO OTRA INFORMACION, ESTAN CONTENIDAS EN EL NIS:
 - v **NETWORK INFORMATION SERVICE: SERVICIO DE INFORMACION DE LA RED.**
 - v ALMACENA PAREJAS (CLAVE, VALOR).
 - v CUANDO SE PROPORCIONA UNA CLAVE, REGRESA EL VALOR CORRESPONDIENTE.
 - v ALMACENA LA ASOCIACION DE:
 - ⊕ LOS NOMBRES DE LOS USUARIOS CON LAS CONTRASEÑAS (CIFRADAS).
 - ⊕ LOS NOMBRES DE LAS MAQUINAS CON LAS DIRECCIONES EN LA RED Y OTROS ELEMENTOS.
- v *IMPLANTACION DE NFS.*
- v LA IMPLANTACION DEL CODIGO DEL CLIENTE Y EL SERVIDOR ES INDEPENDIENTE DE LOS PROTOCOLOS NFS.
- v UNA IMPLEMENTACION QUE SUELE TOMARSE COMO REFERENCIA ES LA DE SUN, QUE CONSTA DE TRES CAPAS.
- v LA **CAPA SUPERIOR** ES LA DE **LLAMADAS AL SISTEMA**:
 - v MANEJA LAS LLAMADAS DEL TIPO OPEN, READ Y CLOSE.
 - v ANALIZA LA LLAMADA Y VERIFICA LOS PARAMETROS.
 - v LLAMA A LA **SEGUNDA CAPA: CAPA DEL SISTEMA VIRTUAL DE ARCHIVOS: VIRTUAL FILE SYSTEM: VFS.**
- v LA CAPA VFS MANTIENE UNA TABLA CON UNA ENTRADA POR CADA ARCHIVO ABIERTO:
 - v ES ANALOGA A LA TABLA DE NODOS-I PARA LOS ARCHIVOS ABIERTOS EN UNIX.

SISTEMAS OPERATIVOS DE REDES

- v LA CAPA VFS TIENE UNA ENTRADA POR CADA ARCHIVO ABIERTO:
 - u SE LA LLAMA **NODO-V (NODO-I VIRTUAL)**.
 - u LOS NODOS-V SE UTILIZAN PARA INDICAR SI EL ARCHIVO ES LOCAL O REMOTO.
 - u PARA LOS ARCHIVOS REMOTOS, POSEEN LA INFORMACION SUFICIENTE COMO PARA TENER ACCESO A ELLOS.
- v PARA MONTAR UN SISTEMA REMOTO DE ARCHIVOS, EL ADMINISTRADOR DEL SISTEMA LLAMA AL PROGRAMA *MOUNT* :
 - u UTILIZA LA INFORMACION DEL DIRECTORIO REMOTO, EL DIRECTORIO LOCAL DONDE SERA MONTADO Y OTROS DATOS ADICIONALES.
 - u CON EL NOMBRE DEL DIRECTORIO REMOTO POR MONTAR SE DESCUBRE EL NOMBRE DE LA MAQUINA DONDE SE LOCALIZA DICHO DIRECTORIO.
 - u SE VERIFICA SI EL DIRECTORIO EXISTE Y SI ESTA DISPONIBLE PARA SU MONTAJE REMOTO.
- v EL NUCLEO:
 - u CONSTRUYE UN **NODO-V** PARA EL DIRECTORIO REMOTO.
 - u PIDE EL CODIGO DEL CLIENTE NFS PARA CREAR UN **NODO-R (NODO-I REMOTO)** EN SUS TABLAS INTERNAS.
- v EL NODO-V APUNTA AL NODO-R.
- v CADA NODO-V DE LA CAPA VFS CONTENDRA EN ULTIMA INSTANCIA UN APUNTADOR A UN NODO-I EN EL S. O. LOCAL.
- v ES POSIBLE VER DESDE EL NODO-V SI UN ARCHIVO O DIRECTORIO ES LOCAL O REMOTO Y, SI ES REMOTO, ENCONTRAR SU ASA DE ARCHIVO.
- v TODO ARCHIVO O DIRECTORIO ABIERTO TIENE UN NODO-V QUE APUNTA A UN NODO-R O A UN NODO-I.
- v POR RAZONES DE EFICIENCIA LAS TRANSFERENCIAS ENTRE CLIENTE Y SERVIDOR SE HACEN EN BLOQUES GRANDES, GENERALMENTE DE 8K:
 - u LUEGO DE HABER RECIBIDO LA CAPA VFS DEL CLIENTE EL BLOQUE NECESARIO, EMITE LA SOLICITUD DEL SIGUIENTE BLOQUE:
 - Φ ESTO SE DENOMINA **LECTURA ADELANTADA (READ AHEAD)**.

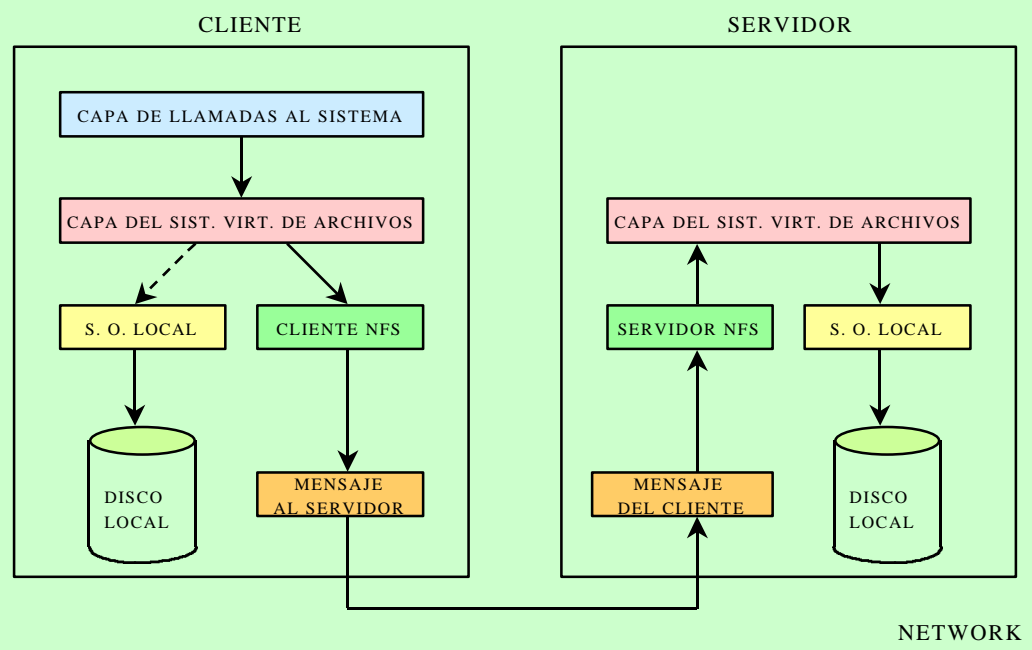
SISTEMAS OPERATIVOS DE REDES

- v UN CRITERIO SIMILAR SE SIGUE CON LA ESCRITURA:
 - u ANTES DE SER ENVIADOS AL SERVIDOR LOS DATOS SE ACUMULAN EN FORMA LOCAL:
 - Φ HASTA COMPLETAR CIERTA CANTIDAD DE BYTES.
 - Φ HASTA QUE SE CIERRA EL ARCHIVO.
- v OTRA TECNICA UTILIZADA PARA MEJORAR EL RENDIMIENTO ES EL **OCULTAMIENTO O CACHING**:
 - u LOS SERVIDORES OCULTAN LOS DATOS PARA EVITAR EL ACCESO AL DISCO:
 - Φ ESTO ES INVISIBLE PARA LOS CLIENTES.
- v LOS CLIENTES MANTIENEN DOS CACHES:
 - u UNO PARA LOS ATRIBUTOS DE ARCHIVO (NODOS-I).
 - u OTRO PARA LOS DATOS DEL ARCHIVO.
- v CUANDO SE NECESITA UN NODO-I O UN BLOQUE DEL ARCHIVO:
 - u PRIMERO SE VERIFICA SI LA SOLICITUD SE PUEDE SATISFACER MEDIANTE EL CACHE DEL CLIENTE:
 - Φ SE EVITA EL TRAFICO EN LA RED.
- v UN PROBLEMA IMPORTANTE DEL CACHING ES QUE EL CACHE NO ES COHERENTE; EJ.:
 - u DOS CLIENTES OCULTAN EL MISMO BLOQUE DEL ARCHIVO.
 - u UNO DE ELLOS LO MODIFICA.
 - u CUANDO EL OTRO LEE EL BLOQUE, OBTIENE EL VALOR ANTIGUO.
 - u PARA MITIGAR ESTE PROBLEMA, LA IMPLANTACION DE NFS:
 - Φ ASOCIA A CADA BLOQUE CACHE UN TEMPORIZADOR (TIMER).
 - Φ CUANDO EL TIMER EXPIRA, LA ENTRADA SE DESCARTA.
 - Φ GENERALMENTE LOS TIEMPOS SON DE:
 - 3 SEG. PARA BLOQUES DE DATOS.
 - 30 SEG. PARA BLOQUES DE DIRECTORIO.

SISTEMAS OPERATIVOS DE REDES

- υ AL ABRIR UN ARCHIVO CON CACHE:
 - Φ SE ENVIA UN MENSAJE AL SERVIDOR PARA REVISAR LA HORA DE LA ULTIMA MODIFICACION.
 - υ SE DETERMINA SI LA COPIA DEL CACHE ES VALIDA O DEBE DESCARTARSE, UTILIZANDO UNA NUEVA COPIA DEL SERVIDOR.
 - υ EL TEMPORIZADOR DEL CACHE EXPIRA CADA 30 SEG.:
 - Φ TODOS LOS BLOQUES MODIFICADOS EN EL CACHE SE ENVIAN AL SERVIDOR.
- v RESUMIENDO:
- υ NFS SOLO TRATA EL SISTEMA DE ARCHIVOS.
 - υ NFS NO HACE REFERENCIA A OTROS ASPECTOS, COMO LA EJECUCION DE UN PROCESO.
 - υ NFS SE HA DIFUNDIDO AMPLIAMENTE, A PESAR DE TODO.

ESTRUCTURA DE LA CAPA NFS



SISTEMAS DISTRIBUIDOS

REALMENTE

- v *NFS ES UN EJEMPLO DE SOFTWARE DEBILMENTE ACOPLADO EN HARDWARE DEBILMENTE ACOPLADO:*
 - u CADA COMPUTADORA PUEDE EJECUTAR SU PROPIO S. O.
 - u SOLO SE DISPONE DE UN SISTEMA COMPARTIDO DE ARCHIVOS.
 - u EL TRAFICO CLIENTE - SERVIDOR DEBE OBEDECER LOS PROTOCOLOS NFS.
- v *LAS MULTICOMPUTADORAS SON UN EJEMPLO DE SOFTWARE FUERTEMENTE ACOPLADO EN HARDWARE DEBILMENTE ACOPLADO:*
 - u CREAN LA ILUSION DE QUE TODA LA RED DE COMPUTADORAS ES UN SOLO SISTEMA DE TIEMPO COMPARTIDO, EN VEZ DE UNA COLECCION DE MAQUINAS DIVERSAS.
- v *UN SISTEMA DISTRIBUIDO ES AQUEL QUE SE EJECUTA EN UNA COLECCION DE MAQUINAS SIN MEMORIA COMPARTIDA, PERO QUE APARECE ANTE SUS USUARIOS COMO UNA SOLA COMPUTADORA:*
 - u A ESTA PROPIEDAD SE LA CONOCE COMO LA **IMAGEN DE UN UNICO SISTEMA.**
- v *TAMBIEN SE DEFINE UN SISTEMA DISTRIBUIDO COMO AQUEL QUE SE EJECUTA EN UNA COLECCION DE MAQUINAS ENLAZADAS MEDIANTE UNA RED PERO QUE ACTUAN COMO UN **UNIPROCESADOR VIRTUAL.***
- v *ALGUNAS DE LAS CARACTERISTICAS DE LOS SISTEMAS DISTRIBUIDOS SON LAS SIGUIENTES:*
 - u DEBE EXISTIR UN MECANISMO DE COMUNICACION GLOBAL ENTRE LOS PROCESOS:
 - Φ CUALQUIER PROCESO DEBE PODER COMUNICARSE (INTERCAMBIAR INFORMACION) CON CUALQUIER OTRO.
 - u NO TIENE QUE HABER:
 - Φ DISTINTOS MECANISMOS EN DISTINTAS MAQUINAS.
 - Φ DISTINTOS MECANISMOS PARA LA COMUNICACION LOCAL O LA COMUNICACION REMOTA.
 - u DEBE EXISTIR UN ESQUEMA GLOBAL DE PROTECCION.
 - u LA ADMINISTRACION DE PROCESOS DEBE SER LA MISMA EN TODAS PARTE.
 - u SE DEBE TENER UNA MISMA INTERFAZ DE LLAMADAS AL SISTEMA EN TODAS PARTES:
 - Φ ES NORMAL QUE SE EJECUTEN NUCLEOS IDENTICOS EN TODAS LAS CPU DEL SISTEMA.
 - u ES NECESARIO UN SISTEMA GLOBAL DE ARCHIVOS.

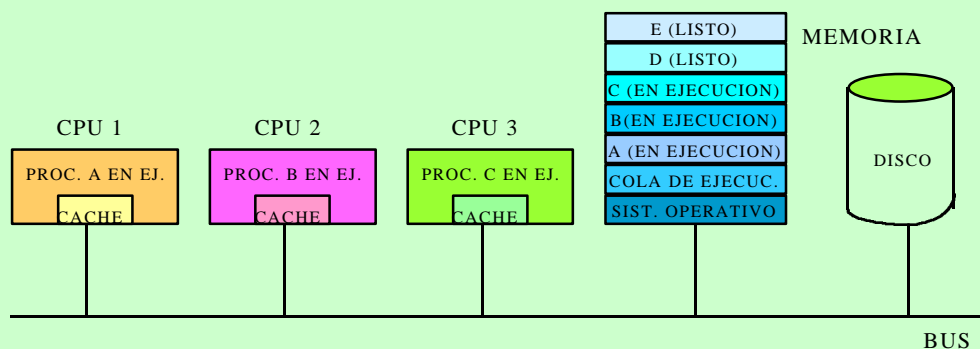
SISTEMAS DE MULTIPROCESADOR CON TIEMPO COMPARTIDO

- v CORRESPONDE A SOFTWARE FUERTEMENTE ACOPLADO EN HARDWARE FUERTEMENTE ACOPLADO.
- v LOS EJEMPLOS MAS COMUNES DE PROPOSITO GENERAL SON LOS MULTIPROCESADORES:
 - u OPERAN COMO UN SISTEMA DE TIEMPO COMPARTIDO, PERO CON VARIAS CPU EN VEZ DE UNA SOLA.
 - u EXTERNAMENTE UN MULTIPROCESADOR CON 32 CPU DE 3 MIPS ACTUA DE MANERA MUY PARECIDA A UNA SOLA CPU DE 96 MIPS:
 - Φ 1 MIPS: 1.000.000 DE INSTRUCCIONES POR SEGUNDO.
 - u SE CORRESPONDE CON LA IMAGEN DE UN UNICO SISTEMA.
- v LA CARACTERISTICA CLAVE ES LA EXISTENCIA DE UNA SOLA COLA PARA EJECUCION:
 - u UNA LISTA DE TODOS LOS PROCESOS EN EL SISTEMA QUE NO ESTAN BLOQUEADOS EN FORMA LOGICA Y LISTOS PARA SU EJECUCION.
 - u LA COLA DE EJECUCION ES UNA ESTRUCTURA DE DATOS CONTENIDA EN LA MEMORIA COMPARTIDA.
- v LOS PROGRAMAS DE LOS PROCESOS ESTAN EN LA MEMORIA COMPARTIDA, TAMBIEN EL S. O.
- v EL PLANIFICADOR (DE PROCESOS) DEL S. O. SE EJECUTA COMO UNA "REGION CRITICA":
 - u SE EVITA QUE DOS CPU ELIJAN EL MISMO PROCESO PARA SU EJECUCION INMEDIATA.
- v CUANDO UN PROCESO SE ASIGNA A UN PROCESADOR:
 - u ENCUENTRA QUE EL CACHE DEL PROCESADOR ESTA OCUPADO POR PALABRAS DE MEMORIA QUE PERTENECEN A AQUELLA PARTE DE LA MEMORIA COMPARTIDA QUE CONTIENE AL PROGRAMA DEL PROCESO ANTERIOR.
 - u LUEGO DE UN BREVE LAPSO SE HABRAN REEMPLAZADO POR EL CODIGO Y LOS DATOS DEL PROGRAMA DEL PROCESO ASIGNADO A ESE PROCESADOR.
- v NINGUNA CPU TIENE MEMORIA LOCAL:
 - u TODOS LOS PROGRAMAS SE ALMACENAN EN LA MEMORIA GLOBAL COMPARTIDA.

SISTEMAS DE MULTIPROCESADOR CON TIEMPO COMPARTIDO

- v SI TODAS LAS CPU ESTAN INACTIVAS EN ESPERA DE E / S Y UN PROCESO ESTA LISTO PARA SU EJECUCION:
 - v ES CONVENIENTE ASIGNARLO A LA CPU QUE SE UTILIZO POR ULTIMA VEZ (PARA ESE PROCESO):
 - ⊕ LA HIPOTESIS ES QUE NINGUN OTRO PROCESO UTILIZO ESA CPU DESDE ENTONCES (HIPOTESIS DE VASWANI Y ZAHORJAN).
- v SI UN PROCESO SE BLOQUEA EN ESPERA DE E / S EN UN MULTIPROCESADOR, EL S. O. PUEDE:
 - v SUSPENDERLO.
 - v DEJARLO EN “ESPERA OCUPADA”:
 - ⊕ ES APLICABLE CUANDO LA MAYORIA DE LA E / S SE REALIZA EN MENOS TIEMPO DEL QUE TARDA UN CAMBIO ENTRE LOS PROCESOS.
 - ⊕ EL PROCESO CONSERVA SU PROCESADOR POR ALGUNOS MILISEGUNDOS EN ESPERA DE QUE LA E / S FINALICE:
 - SI SE AGOTA EL TIEMPO DE ESPERA Y NO HA FINALIZADO LA E / S, SE REALIZA UNA CONMUTACION DE PROCESOS.
- v GENERALMENTE SE DISPONDRÁ DE UN SISTEMA DE ARCHIVOS TRADICIONAL, CON UN UNICO CACHE:
 - v GLOBALMENTE CONSIDERADO ES SIMILAR AL SISTEMA DE ARCHIVOS DE UN UNICO PROCESADOR.

UN MULTIPROCESADOR CON UNA SOLA COLA DE EJECUCION



ASPECTOS DEL DISEÑO

- v LA COMPARACION DE LAS TRES PRINCIPALES FORMAS DE ORGANIZAR N CPU SE PUEDE RESUMIR EN EL SIGUIENTE CUADRO:

<u>ELEMENTO</u>	<u>S. O. DE RED</u>	<u>S. O. DISTRIB.</u>	<u>S. O. DE MULTIPR.</u>
¿SE VE COMO UN UNIPRO- CESADOR VIRTUAL?	NO	SI	SI
¿TODAS TIENEN QUE EJE- CUTAR EL MISMO S. O.?	NO	SI	SI
¿CUANTAS COPIAS DEL S. O. EXISTEN?	N	N	1
¿COMO SE LOGRA LA COMUNICACION?	ARCHIVOS COMPARTIDOS	MENSAJES	MEMORIA COMPARTIDA
¿SE REQUIERE UN ACUERDO EN LOS PRO- TOCOLOS DE LA RED?	SI	SI	NO
¿EXISTE UNA UNICA COLA DE EJECUCION?	NO	NO	SI
¿EXISTE UNA SEMANTICA BIEN DEFINIDA PARA LOS ARCHIVOS COMPARTIDOS?	POR LO GENERAL NO	SI	SI

- v LOS ASPECTOS CLAVES EN EL DISEÑO DE S. O. DISTRIBUIDOS SON:
 - v TRANSPARENCIA, FLEXIBILIDAD, CONFIABILIDAD, DESEMPEÑO Y ESCALABILIDAD.

TRANSPARENCIA

- v UN ASPECTO MUY IMPORTANTE ES LA FORMA DE LOGRAR LA IMAGEN DE UN UNICO SISTEMA.
- v LOS USUARIOS DEBEN PERCIBIR QUE LA COLECCION DE MAQUINAS CONECTADAS SON UN SISTEMA DE TIEMPO COMPARTIDO DE UN SOLO PROCESADOR:
 - u UN SISTEMA QUE LOGRE ESTE OBJETIVO SE DICE QUE ES **TRANSPARENTE**.
- v *DESDE EL PUNTO DE VISTA DE LOS USUARIOS*, LA TRANSPARENCIA SE LOGRA CUANDO:
 - u SUS PEDIDOS SE SATISFACEN CON EJECUCIONES EN PARALELO EN DISTINTAS MAQUINAS.
 - u SE UTILIZAN UNA VARIEDAD DE SERVIDORES DE ARCHIVOS.
 - u EL USUARIO NO NECESITA SABERLO NI NOTARLO.
- v LA TRANSPARENCIA *DESDE EL PUNTO DE VISTA DE LOS PROGRAMAS* SIGNIFICA DISEÑAR LA INTERFAZ DE LLAMADAS AL SISTEMA DE MODO QUE NO SEA VISIBLE LA EXISTENCIA DE VARIOS PROCESADORES.
- v NO ES TRANSPARENTE UN SISTEMA DONDE EL ACCESO A LOS ARCHIVOS REMOTOS SE REALICE MEDIANTE:
 - u EL ESTABLECIMIENTO EXPLICITO DE UNA CONEXION EN LA RED CON UN SERVIDOR REMOTO.
 - u EL ENVIO POSTERIOR DE MENSAJES:
 - Φ EL ACCESO A LOS SERVICIOS REMOTOS SERA DISTINTO AL ACCESO A LOS SERVICIOS LOCALES.
- v EXISTEN DISTINTOS TIPOS DE **TRANSPARENCIA** EN UN SISTEMA DISTRIBUIDO:
 - u **DE LOCALIZACION:** LOS USUARIOS NO PUEDEN INDICAR LA LOCALIZACION DE LOS RECURSOS.
 - u **DE MIGRACION:** LOS RECURSOS SE PUEDEN MOVER A VOLUNTAD SIN CAMBIAR SUS NOMBRES.
 - u **DE REPLICA:** LOS USUARIOS NO PUEDEN INDICAR EL N° DE COPIAS EXISTENTES.
 - u **DE CONCURRENCIA:** VARIOS USUARIOS PUEDEN COMPARTIR RECURSOS DE MANERA AUTOMATICA.
 - u **DE PARALELISMO:** LAS ACTIVIDADES PUEDEN OCURRIR EN PARALELO SIN EL CONOCIMIENTO DE LOS USUARIOS.

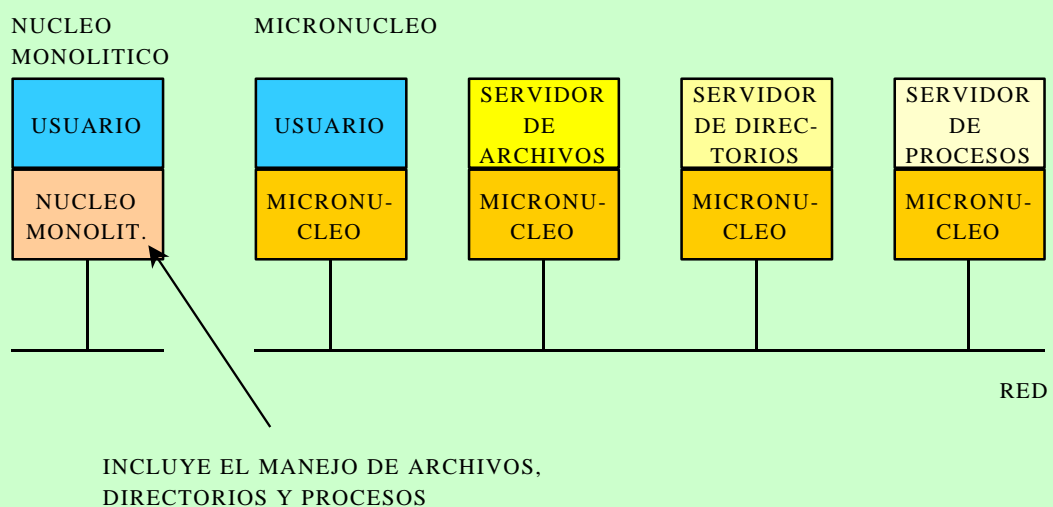
FLEXIBILIDAD

- v LA FLEXIBILIDAD ES DE FUNDAMENTAL IMPORTANCIA.
- v EXISTEN DOS ESCUELAS DE PENSAMIENTO EN CUANTO A LA ESTRUCTURA DE LOS SISTEMAS DISTRIBUIDOS:
 - v **NUCLEO MONOLITICO:**
 - Φ CADA MAQUINA DEBE EJECUTAR UN NUCLEO TRADICIONAL QUE PROPORCIONE LA MAYORIA DE LOS SERVICIOS.
 - v **MICRONUCLEO (MICROKERNEL):**
 - Φ EL NUCLEO DEBE PROPORCIONAR LO MENOS POSIBLE.
 - Φ EL GRUESO DE LOS SERVICIOS DEL S. O. SE DEBE OBTENER A PARTIR DE LOS SERVIDORES AL NIVEL USUARIO.
- v EL NUCLEO MONOLITICO ES EL S. O. CENTRALIZADO AUMENTADO CON:
 - v CAPACIDADES DE RED.
 - v INTEGRACION DE SERVICIOS REMOTOS.
- v CON NUCLEO MONOLITICO:
 - v LA MAYORIA DE LAS LLAMADAS AL SISTEMA SE REALIZAN MEDIANTE SEÑALAMIENTO AL NUCLEO:
 - Φ EL NUCLEO REALIZA EL TRABAJO.
 - Φ EL NUCLEO REGRESA EL RESULTADO AL PROCESO DEL USUARIO.
 - v LA MAYORIA DE LAS MAQUINAS TIENE DISCOS Y ADMINISTRA SUS PROPIOS SISTEMAS LOCALES DE ARCHIVOS.
- v EL MICRONUCLEO ES MAS FLEXIBLE Y PROPORCIONA SOLO CUATRO SERVICIOS MINIMOS:
 - v UN MECANISMO DE COMUNICACION ENTRE PROCESOS.
 - v CIERTA ADMINISTRACION DE LA MEMORIA.
 - v UNA CANTIDAD LIMITADA DE PLANIFICACION Y ADMINISTRACION DE PROCESOS DE BAJO NIVEL.
 - v ENTRADA / SALIDA DE BAJO NIVEL.
- v CONTRARIAMENTE AL NUCLEO MONOLITICO, EL MICRONUCLEO NO PROPORCIONA:
 - v EL SISTEMA DE ARCHIVOS, EL SISTEMA DE DIRECTORIOS, TODA LA ADMINISTRACION DE PROCESOS O GRAN PARTE DEL MANEJO DE LAS LLAMADAS AL SISTEMA.

FLEXIBILIDAD

- v EL OBJETIVO ES MANTENER EL MICRONUCLEO PEQUEÑO.
- v TODOS LOS DEMAS SERVICIOS DEL S. O. SE IMPLEMENTAN GENERALMENTE COMO SERVIDORES A NIVEL USUARIO:
 - v PARA OBTENER UN SERVICIO:
 - Φ EL USUARIO ENVIA UN MENSAJE AL SERVIDOR APROPIADO.
 - Φ EL SERVIDOR REALIZA EL TRABAJO Y REGRESA EL RESULTADO.
- v UNA IMPORTANTE VENTAJA DE ESTE METODO ES SU ALTA MODULARIDAD:
 - v EXISTE UNA INTERFAZ BIEN DEFINIDA CON CADA SERVICIO (CONJUNTO DE MENSAJES QUE COMPRENDE EL SERVIDOR).
 - v CADA SERVICIO ES IGUAL DE ACCESIBLE PARA TODOS LOS CLIENTES, INDEPENDIENTEMENTE DE LA POSICION.
 - v ES FACIL IMPLANTAR, INSTALAR Y DEPURAR NUEVOS SERVICIOS, SIN NECESIDAD DE DETENER EL SISTEMA TOTALMENTE.

ESQUEMA DE NUCLEO MONOLITICO Y MICRONUCLEO



CONFIABILIDAD

- v UN IMPORTANTE OBJETIVO DE LOS SISTEMAS DISTRIBUIDOS ES QUE SI UNA MAQUINA FALLA, ALGUNA OTRA DEBE ENCARGARSE DEL TRABAJO.
- v LA CONFIABILIDAD GLOBAL *TEORICA* DEL SISTEMA PODRIA SER EL “OR” BOOLEANO DE LA CONFIABILIDAD DE LOS COMPONENTES; EJEMPLO:
 - v SE DISPONE DE 5 SERVIDORES DE ARCHIVOS, C / U CON UNA PROBABILIDAD DE 0,95 DE FUNCIONAR EN UN INSTANTE DADO.
 - v LA PROBABILIDAD DE FALLA SIMULTANEA DE LOS 5 ES $(0,05)^5 = 0,000006$.
 - v LA PROBABILIDAD DE QUE AL MENOS UNO ESTE DISPONIBLE ES 0,999994.
- v LA CONFIABILIDAD *PRACTICA* SE VE DISMINUIDA YA QUE MUCHAS VECES SE REQUIERE QUE CIERTOS SERVIDORES ESTEN EN SERVICIO SIMULTANEAMENTE PARA QUE EL TODO FUNCIONE:
 - v ALGUNOS SISTEMAS TIENEN UNA DISPONIBILIDAD MAS RELACIONADA CON EL “AND” BOOLEANO DE LAS COMPONENTES QUE CON EL “OR” BOOLEANO.
- v UN ASPECTO DE LA CONFIABILIDAD ES LA **DISPONIBILIDAD**:
 - v SE REFIERE A LA FRACCION DE TIEMPO EN QUE SE PUEDE UTILIZAR EL SISTEMA.
- v LA DISPONIBILIDAD SE MEJORA MEDIANTE:
 - v UN DISEÑO QUE NO EXIJA EL FUNCIONAMIENTO SIMULTANEO DE UN N° SUSTANCIAL DE COMPONENTES CRITICOS.
 - v LA REDUNDANCIA, ES DECIR LA DUPLICIDAD DE COMPONENTES CLAVE DEL HARDWARE Y DEL SOFTWARE.
- v LOS DATOS NO DEBEN PERDERSE O MEZCLARSE:
 - v SI LOS ARCHIVOS SE ALMACENAN DE MANERA REDUNDANTE EN VARIOS SERVIDORES, TODAS LAS COPIAS DEBEN SER CONSISTENTES.
- v OTRO ASPECTO DE LA CONFIABILIDAD GENERAL ES LA **SEGURIDAD**:
 - v LOS ARCHIVOS Y OTROS RECURSOS DEBEN SER PROTEGIDOS CONTRA EL USO NO AUTORIZADO.
- v OTRO ASPECTO RELACIONADO CON LA CONFIABILIDAD ES LA **TOLERANCIA A FALLAS**:
 - v LAS FALLAS SE DEBEN OCULTAR BRINDANDO UNA RECUPERACION TRANSPARENTE PARA EL USUARIO, AUNQUE HAYA CIERTA DEGRADACION DE LA PERFORMANCE.

DESEMPEÑO

- v CUANDO SE EJECUTA UNA APLICACION EN UN SISTEMA DISTRIBUIDO NO DEBE PARECER PEOR QUE SU EJECUCION EN UN UNICO PROCESADOR:
 - u ESTO ES DIFICIL DE LOGRAR.
- v ALGUNAS METRICAS DEL DESEMPEÑO SON:
 - u TIEMPO DE RESPUESTA.
 - u RENDIMIENTO (N° DE TRABAJOS POR HORA).
 - u USO DEL SISTEMA Y CANTIDAD CONSUMIDA DE LA CAPACIDAD DE LA RED.
- v EL PROBLEMA SE COMPLICA POR EL HECHO DE QUE LA COMUNICACION ENTRE EQUIPOS ES LENTA COMPARADA CON:
 - u LA VELOCIDAD DE PROCESO.
 - u LA VELOCIDAD DE LA COMUNICACION DENTRO DE UN MISMO PROCESADOR.
- v SE REQUIERE EL USO DE PROTOCOLOS DE COMUNICACIONES EN LOS EXTREMOS (PROCESADORES) QUE INTERVIENEN EN LA COMUNICACION:
 - u SE INCREMENTA EL CONSUMO DE CICLOS DE PROCESADOR.
- v PARA OPTIMIZAR EL DESEMPEÑO FRECUENTEMENTE HAY QUE:
 - u MINIMIZAR EL N° DE MENSAJES:
 - Φ LA DIFICULTAD ES QUE LA MEJOR FORMA DE MEJORAR EL DESEMPEÑO ES TENER MUCHAS ACTIVIDADES EN EJECUCION PARALELA EN DISTINTOS PROCESADORES:
 - ESTO REQUIERE EL ENVIO DE MUCHOS MENSAJES.
 - u CENTRALIZAR EL TRABAJO EN UNA SOLA MAQUINA:
 - Φ RESULTA POCO APROPIADO PARA UN SISTEMA DISTRIBUIDO.
- v TAMBIEN SE DEBE PRESTAR ATENCION AL **TAMAÑO DE GRANO** DE TODOS LOS CALCULOS:
 - u **PARALELISMO DE GRANO FINO:**
 - Φ CORRESPONDE A TRABAJOS CON UN GRAN N° DE PEQUEÑOS CALCULOS Y MUCHA INTERACCION CON OTROS TRABAJOS:
 - REQUIEREN MUCHA COMUNICACION QUE PUEDE AFECTAR EL DESEMPEÑO.
 - u **PARALELISMO DE GRANO GRUESO:**
 - Φ CORRESPONDE A TRABAJOS CON GRANDES CALCULOS, POCA INTERACCION Y POCOS DATOS:
 - REQUIEREN POCA COMUNICACION Y NO AFECTAN LA PERFORMANCE.

ESCALABILIDAD

- v LA TENDENCIA INDICA QUE EL TAMAÑO DE LOS SISTEMAS DISTRIBUIDOS ES HACIA CIENTOS DE MILES Y AUN DECENAS DE MILLONES DE USUARIOS CONECTADOS.
- v EXISTEN CUELLOS DE BOTELLA POTENCIALES QUE SE DEBE INTENTAR EVITAR EN LOS SISTEMAS DISTRIBUIDOS DE GRAN ESCALA:
 - v COMPONENTES CENTRALIZADOS:
 - Φ EJ.: UN SOLO SERVIDOR DE CORREO PARA TODOS LOS USUARIOS.
 - v TABLAS CENTRALIZADAS:
 - Φ EJ.: UN UNICO DIRECTORIO TELEFONICO EN LINEA.
 - v ALGORITMOS CENTRALIZADOS:
 - Φ EJ.: REALIZACION DE UN RUTEO CON BASE EN LA INFORMACION COMPLETA.
- v SE DEBEN UTILIZAR ALGORITMOS DESCENTRALIZADOS CON LAS SIGUIENTES CARACTERISTICAS:
 - v NINGUNA MAQUINA TIENE LA INFORMACION COMPLETA ACERCA DEL ESTADO DEL SISTEMA.
 - v LAS MAQUINAS TOMAN DECISIONES SOLO EN BASE A LA INFORMACION DISPONIBLE DE MANERA LOCAL.
 - v EL FALLO DE UNA MAQUINA NO ARRUINA EL ALGORITMO.
 - v NO EXISTE UNA HIPOTESIS IMPLICITA DE LA EXISTENCIA DE UN RELOJ GLOBAL.

COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS

- v INTRODUCCION A LA COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS
- v PROTOCOLOS CON CAPAS
- v INTRODUCCION AL MODELO CLIENTE - SERVIDOR (C - S)
- v DIRECCIONAMIENTO EN C - S
- v PRIMITIVAS DE BLOQUEO VS. NO BLOQUEO EN C - S
- v PRIMITIVAS ALMACENADAS EN BUFFER VS. NO ALMACENADAS EN C - S
- v PRIMITIVAS CONFIABLES VS. NO CONFIABLES EN C - S
- v IMPLANTACION DEL MODELO C - S
- v LLAMADA A UN PROCEDIMIENTO REMOTO (RPC)
- v OPERACION BASICA DE RPC
- v TRANSFERENCIA DE PARAMETROS EN RPC
- v CONEXION DINAMICA EN RPC
- v SEMANTICA DE RPC EN PRESENCIA DE FALLOS
- v ASPECTOS DE LA IMPLANTACION DE RPC
- v AREAS DE PROBLEMAS EN RPC
- v COMUNICACION EN GRUPO
- v ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

INTRODUCCION A LA COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS

- v LA DIFERENCIA MAS IMPORTANTE ENTRE UN SISTEMA DISTRIBUIDO Y UN SISTEMA DE UN UNICO PROCESADOR ES LA COMUNICACION ENTRE PROCESOS.
- v EN UN SISTEMA DE UN SOLO PROCESADOR LA COMUNICACION SUPONE IMPLICITAMENTE LA EXISTENCIA DE LA MEMORIA COMPARTIDA:
 - u EJ.: PROBLEMA DE LOS PRODUCTORES Y LOS CONSUMIDORES:
 - Φ UN PROCESO ESCRIBE EN UN BUFFER COMPARTIDO Y OTRO PROCESO LEE DE EL.
- v EN UN SISTEMA DISTRIBUIDO NO EXISTE LA MEMORIA COMPARTIDA:
 - u TODA LA NATURALEZA DE LA COMUNICACION ENTRE PROCESOS DEBE REPLANTEARSE.
- v LOS PROCESOS, PARA COMUNICARSE, DEBEN APEGARSE A REGLAS CONOCIDAS COMO *PROTOCOLOS*.
- v PARA LOS SISTEMAS DISTRIBUIDOS EN UN AREA AMPLIA, ESTOS PROTOCOLOS TOMAN FRECUENTEMENTE LA FORMA DE VARIAS CAPAS:
 - u CADA CAPA TIENE SUS PROPIAS METAS Y REGLAS.
- v LOS MENSAJES SE INTERCAMBIAN DE DIVERSAS FORMAS, EXISTIENDO MUCHAS OPCIONES DE DISEÑO AL RESPECTO:
 - u UNA IMPORTANTE OPCION ES LA LLAMADA A UN PROCEDIMIENTO REMOTO.
- v TAMBIEN ES IMPORTANTE CONSIDERAR LAS POSIBILIDADES DE COMUNICACION ENTRE GRUPOS DE PROCESOS, NO SOLO ENTRE DOS PROCESOS.

PROTOCOLOS CON CAPAS

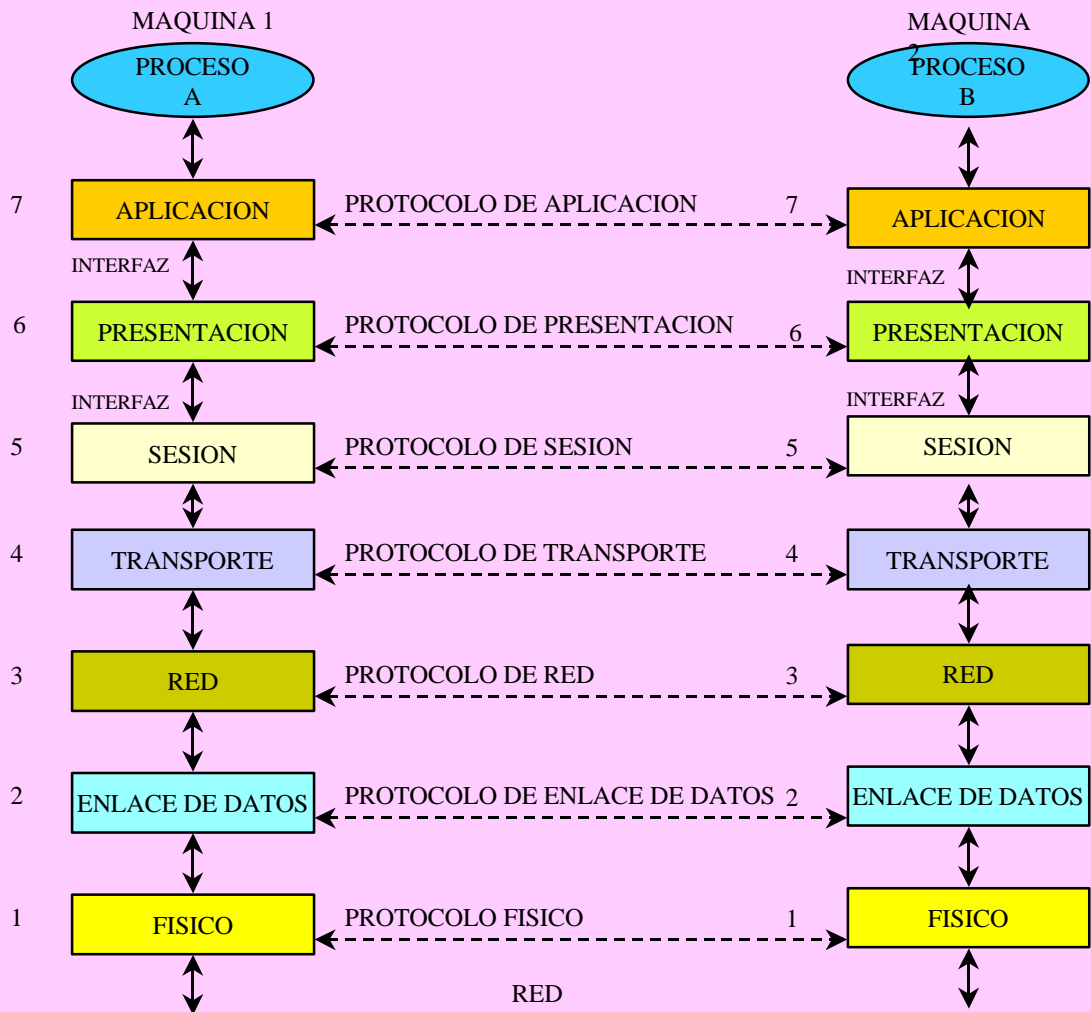
- v DEBIDO A LA AUSENCIA DE MEMORIA COMPARTIDA, TODA LA COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS SE BASA EN LA *TRANSFERENCIA DE MENSAJES*.
- v CUANDO EL PROCESO A QUIERE COMUNICARSE CON EL PROCESO B:
 - u CONSTRUYE UN MENSAJE EN SU PROPIO ESPACIO DE DIRECCIONES.
 - u EJECUTA UNA LLAMADA AL SISTEMA PARA QUE EL S. O. BUSQUE EL MENSAJE Y LO ENVIE A TRAVES DE LA RED HACIA B.
 - u PARA EVITAR EL CAOS, A Y B DEBEN COINCIDIR EN EL SIGNIFICADO DE LOS BITS QUE SE ENVÍEN.
- v LOS PUNTOS DE ACUERDO NECESARIOS INCLUYEN LO SIGUIENTE:
 - u ¿CUANTOS VOLTIOS HAY QUE UTILIZAR PARA UN BIT 0 Y CUANTOS PARA UN BIT 1?.
 - u ¿COMO SABE EL RECEPTOR CUAL ES EL ULTIMO BIT DEL MENSAJE?.
 - u ¿COMO PUEDE DETECTAR SI UN MENSAJE HA SIDO DAÑADO O PERDIDO, Y QUE DEBE HACER SI LO DESCUBRE?.
 - u ¿QUE LONGITUD TIENEN LOS NUMEROS, CADENAS Y OTROS ELEMENTOS DE DATOS Y CUAL ES LA FORMA EN QUE ESTAN REPRESENTADOS?.
- v LA **ISO** (ORGANIZACION INTERNACIONAL DE ESTANDARES) DESARROLLO UN **MODELO DE REFERENCIA** QUE:
 - u IDENTIFICA EN FORMA CLARA LOS DISTINTOS NIVELES.
 - u ESTANDARIZA LOS NOMBRES DE LOS NIVELES.
 - u SEÑALA CUAL NIVEL DEBE REALIZAR CUAL TRABAJO.
- v ESTE MODELO SE DENOMINA **MODELO DE REFERENCIA PARA INTERCONEXION DE SISTEMAS ABIERTOS (ISO OSI O MODELO OSI)**.
- v EL MODELO OSI ESTA DISEÑADO PARA PERMITIR LA COMUNICACION DE LOS **SISTEMAS ABIERTOS**:
 - u SON AQUELLOS PREPARADOS PARA COMUNICARSE CON CUALQUIER OTRO SISTEMA ABIERTO MEDIANTE *REGLAS ESTANDAR*:
 - Φ ESTABLECEN EL FORMATO, CONTENIDO Y SIGNIFICADO DE LOS MENSAJES RECIBIDOS Y ENVIADOS.
 - Φ CONSTITUYEN LOS **PROTOCOLOS**:
 - SON ACUERDOS EN LA FORMA EN QUE DEBE DESARROLLARSE LA COMUNICACION.

PROTOCOLOS CON CAPAS

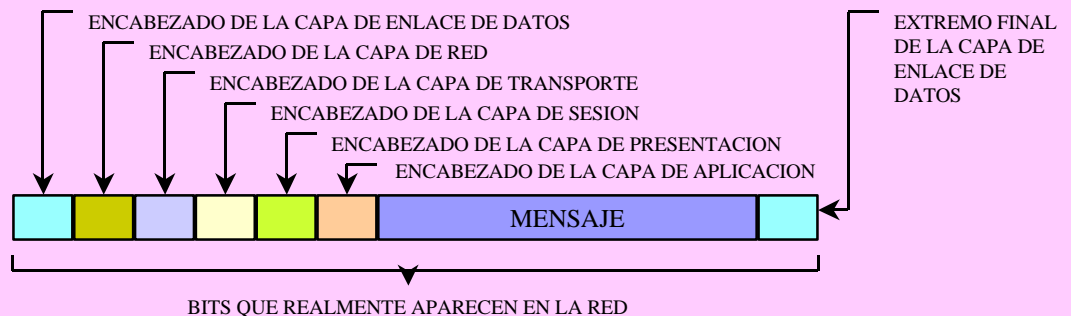
- v EL MODELO OSI DISTINGUE ENTRE DOS TIPOS GENERALES DE **PROTOCOLOS**:
 - u **ORIENTADOS HACIA LAS CONEXIONES:**
 - Φ ANTES DE INTERCAMBIAR LOS DATOS, EL EMISOR Y EL RECEPTOR:
 - ESTABLECEN EN FORMA EXPLICITA UNA CONEXION.
 - PROBABLEMENTE NEGOCIEN EL PROTOCOLO A UTILIZAR.
 - AL FINALIZAR, DEBEN TERMINAR LA CONEXION.
 - EL TELEFONO ES UN SISTEMA DE COMUNICACION ORIENTADO HACIA LA CONEXION.
 - u **SIN CONEXION:**
 - Φ NO ES NECESARIA UNA CONFIGURACION DE ANTEMANO.
 - Φ EL EMISOR TRANSMITE EL PRIMER MENSAJE CUANDO ESTA LISTO.
 - Φ EL DEPOSITO DE UNA CARTA EN UN BUZON ES UNA COMUNICACION SIN CONEXION.
- v CADA CAPA PROPORCIONA UNA **INTERFAZ** CON LA OTRA CAPA POR ENCIMA DE ELLA:
 - u LA INTERFAZ CONSISTE DE UN CONJUNTO DE OPERACIONES PARA DEFINIR EL SERVICIO QUE LA CAPA ESTA PREPARADA PARA OFRECER A SUS USUARIOS.
- v EL PROTOCOLO DE LA CAPA n UTILIZA LA INFORMACION DE LA CAPA n .
- v CADA PROTOCOLO DE CAPA SE PUEDE CAMBIAR INDEPENDIENTEMENTE DE LOS DEMAS:
 - u ESTO ES DE FUNDAMENTAL IMPORTANCIA.
 - u CONFIERE GRAN FLEXIBILIDAD.
- v LA COLECCION DE PROTOCOLOS UTILIZADOS EN UN SISTEMA PARTICULAR SE LLAMA UNA **SUITE DE PROTOCOLO O PILA DE PROTOCOLO**.

PROTOCOLOS CON CAPAS

CAPAS, INTERFACES Y PROTOCOLOS EN EL MODELO OSI



UN MENSAJE TIPO TAL COMO APARECE EN LA RED



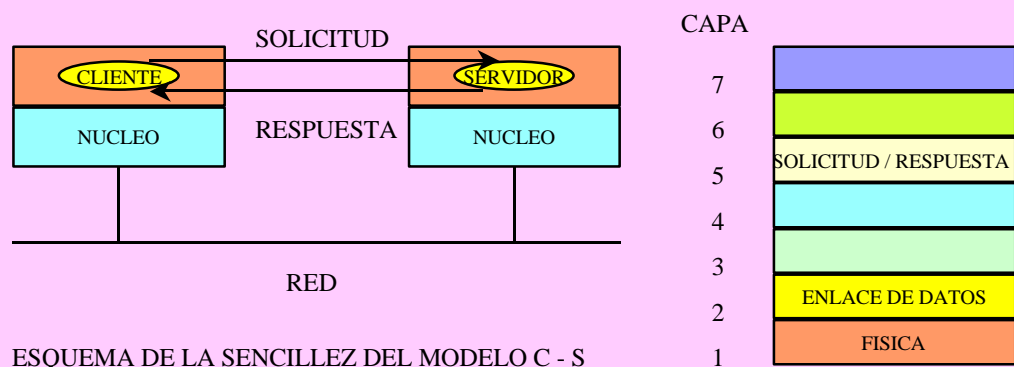
INTRODUCCION AL MODELO CLIENTE - SERVIDOR (C - S)

- v EL MODELO DE LA OSI ES UNA SOLUCION ELEGANTE Y REALMENTE APLICABLE EN MUCHOS CASOS, PERO TIENE UN PROBLEMA:
 - u LA EXISTENCIA DE LOS ENCABEZADOS GENERA UN “COSTO” ADICIONAL DE TRANSMISION.
 - u CADA ENVIO DE UN MENSAJE GENERA:
 - Φ PROCESO EN MEDIA DOCENA DE CAPAS.
 - Φ GENERACION Y AGREGADO DE ENCABEZADOS EN EL CAMINO HACIA “ABAJO”.
 - Φ ELIMINACION Y EXAMEN DE ENCABEZADOS EN EL CAMINO HACIA “ARRIBA”.
- v CON ENLACES DEL ORDEN DE DECENAS (O CENTENAS) DE MILES DE BITS / SEG. Y CPU PODEROSAS:
 - u LA CARGA DE PROCESAMIENTO DE LOS PROTOCOLOS NO ES SIGNIFICATIVA.
 - u EL FACTOR LIMITANTE ES LA CAPACIDAD DE LAS LINEAS.
 - u EJ.: REDES DE AREA EXTENDIDA (WAN).
- v CON ENLACES DEL ORDEN DE MILLONES DE BITS / SEG. Y COMPUTADORAS PERSONALES:
 - u LA CARGA DE PROCESAMIENTO DE LOS PROTOCOLOS SI ES FRECUENTEMENTE SIGNIFICATIVA.
 - u EL FACTOR LIMITANTE NO ES LA CAPACIDAD DE LAS LINEAS.
 - u EJ.: REDES DE AREA LOCAL (LAN).
- v LA MAYORIA DE LOS SISTEMAS DISTRIBUIDOS BASADOS EN LAN NO UTILIZAN LOS PROTOCOLOS DE CAPAS COMPLETOS:
 - u UTILIZAN UN SUBCONJUNTO DE TODA UNA PILA DE PROTOCOLOS.
- v EL *MODELO OSI* NO DICE NADA ACERCA DE LA FORMA DE ESTRUCTURAR AL SISTEMA DISTRIBUIDO.
- v EL *MODELO CLIENTE - SERVIDOR* TIENE COMO IDEA FUNDAMENTAL LA ESTRUCTURACION DEL S. O. COMO:
 - u UN GRUPO DE PROCESOS EN COOPERACION, LLAMADOS **SERVIDORES**, QUE OFRECEN SERVICIOS A LOS USUARIOS.
 - u UN GRUPO DE PROCESOS USUARIOS LLAMADOS **CLIENTES**.

INTRODUCCION AL MODELO CLIENTE - SERVIDOR (C - S)

- v EL MODELO CLIENTE - SERVIDOR SE BASA EN UN **PROTOCOLO SOLICITUD / RESPUESTA**:
 - v ES SENCILLO Y SIN CONEXION.
 - v NO ES COMPLEJO Y ORIENTADO A LA CONEXION COMO OSI O TCP / IP.
 - v EL CLIENTE ENVIA UN MENSAJE DE SOLICITUD AL SERVIDOR PIDIENDO CIERTO SERVICIO.
 - v EL SERVIDOR:
 - Φ EJECUTA EL REQUERIMIENTO.
 - Φ REGRESA LOS DATOS SOLICITADOS O UN CODIGO DE ERROR SI NO PUDO EJECUTARLO CORRECTAMENTE.
 - v NO SE TIENE QUE ESTABLECER UNA CONEXION SINO HASTA QUE ESTA SE UTILICE.
 - v LA PILA DEL PROTOCOLO ES MAS CORTA Y POR LO TANTO MAS EFICIENTE.
 - v SI TODAS LAS MAQUINAS FUESEN IDENTICAS SOLO SE NECESITARIAN TRES NIVELES DE PROTOCOLOS.

MODELO CLIENTE - SERVIDOR



NIVELES DE PROTOCOLOS
NECESARIOS SI TODAS LAS
MAQUINAS FUESEN IDENTICAS

DIRECCIONAMIENTO EN C - S

- v PARA QUE UN CLIENTE PUEDA ENVIAR UN MENSAJE A UN SERVIDOR, DEBE CONOCER LA DIRECCION DE ESTE.
- v UN ESQUEMA DE DIRECCIONAMIENTO SE BASA EN LA *DIRECCION DE LA MAQUINA* DESTINATARIA DEL MENSAJE:
 - u ES LIMITATIVO SI EN LA MAQUINA DESTINATARIA SE EJECUTAN VARIOS PROCESOS:
 - Φ NO SE SABRIA PARA CUAL DE ELLOS ES EL MENSAJE.
- v OTRO ESQUEMA DE DIRECCIONAMIENTO SE BASA EN *IDENTIFICAR LOS PROCESOS* DESTINATARIOS EN VEZ DE A LAS MAQUINAS:
 - u ELIMINA LA AMBIGÜEDAD ACERCA DE QUIEN ES EL RECEPTOR.
 - u PRESENTA EL PROBLEMA DE COMO IDENTIFICAR LOS PROCESOS:
 - Φ UNA SOLUCION ES UNA NOMENCLATURA QUE INCLUYA LA IDENTIFICACION DE LA MAQUINA Y DEL PROCESO:
 - NO SE NECESITAN COORDENADAS GLOBALES.
 - PUEDEN REPETIRSE LOS NOMBRES DE LOS PROCESOS EN DISTINTAS MAQUINAS.
- v UNA VARIANTE UTILIZA *machine.local-id* EN VEZ DE *machine.process*:
 - u *local-id* GENERALMENTE ES UN ENTERO ALEATORIO DE 16 O 32 BITS.
 - u UN PROCESO SERVIDOR SE INICIA MEDIANTE UNA LLAMADA AL SISTEMA PARA INDICARLE AL NUCLEO QUE DESEA ESCUCHAR A *local-id*.
 - u CUANDO SE ENVIA UN MENSAJE DIRIGIDO A *machine.local-id* EL NUCLEO SABE A CUAL PROCESO DEBE DAR EL MENSAJE.
- v EL DIRECCIONAMIENTO *machine.process* PRESENTA EL SERIO INCONVENIENTE DE QUE NO ES TRANSPARENTE:
 - u LA TRANSPARENCIA ES UNO DE LOS PRINCIPALES OBJETIVOS DE LA CONSTRUCCION DE SISTEMAS DISTRIBUIDOS.
 - u EL USUARIO DEBE CONOCER LA POSICION DEL SERVIDOR.
 - u UN CAMBIO DE SERVIDOR OBLIGA A CAMBIAR LOS PROGRAMAS.

DIRECCIONAMIENTO EN C - S

- v OTRO METODO DE DIRECCIONAMIENTO CONSISTE EN ASIGNARLE A CADA PROCESO UNA UNICA DIRECCION QUE NO CONTenga UN N° DE MAQUINA:
 - u UNA FORMA ES MEDIANTE UN ASIGNADOR CENTRALIZADO DE DIRECCIONES A LOS PROCESOS QUE MANTENGA UN CONTADOR:
 - Φ AL RECIBIR UNA SOLICITUD DE DIRECCION REGRESA EL VALOR ACTUAL DEL CONTADOR Y LO INCREMENTA EN UNO.
 - u LA DESVENTAJA ES EL ELEMENTO CENTRALIZADO.
- v TAMBIEN EXISTE EL METODO DE DEJAR QUE CADA PROCESO ELIJA SU PROPIO IDENTIFICADOR:
 - u EN UN ESPACIO DE DIRECCIONES GRANDE Y DISPERSO, POR EJ.: ENTEROS BINARIOS DE 64 BITS.
 - u LA PROBABILIDAD DE QUE DOS PROCESOS ELIJAN EL MISMO N° ES MUY PEQUEÑA.
 - u EXISTE EL PROBLEMA, PARA EL NUCLEO EMISOR, DE SABER A QUE MAQUINA ENVIAR EL MENSAJE:
 - Φ EN UNA LAN, EL EMISOR PUEDE TRANSMITIR UN **PAQUETE ESPECIAL DE LOCALIZACION** CON LA DIRECCION DEL PROCESO DESTINO.
 - Φ ESTE PAQUETE DE TRANSMISION SERA RECIBIDO POR TODAS LAS MAQUINAS DE LA RED.
 - Φ TODOS LOS NUCLEOS VERIFICAN SI LA DIRECCION ES LA SUYA:
 - SI LO ES, REGRESA UN MENSAJE **AQUI ESTOY** CON SU DIRECCION EN LA RED (N° DE MAQUINA).
 - Φ EL NUCLEO EMISOR UTILIZA ESA DIRECCION Y LA CAPTURA PARA EVITAR A POSTERIORI UNA NUEVA BUSQUEDA DEL SERVIDOR.
 - u ES UN ESQUEMA TRANSPARENTE, PERO LA TRANSMISION PROVOCA UNA CARGA ADICIONAL EN EL SISTEMA:
 - Φ SE PUEDE EVITAR CON UNA MAQUINA ADICIONAL PARA LA ASOCIACION DE:
 - LOS NOMBRES DE SERVICIOS.
 - LAS DIRECCIONES DE LAS MAQUINAS.

DIRECCIONAMIENTO EN C - S

- v AL UTILIZAR ESTE SISTEMA:
 - u SE HACE REFERENCIA A LOS PROCESOS DE LOS SERVIDORES MEDIANTE CADENAS EN ASCII:
 - Φ SON LAS QUE APARECEN EN LOS PROGRAMAS.
 - u NO SE REFERENCIAN N° BINARIOS DE MAQUINAS O PROCESOS.
 - u AL EJECUTAR UN CLIENTE QUE INTENTE UTILIZAR UN SERVIDOR:
 - Φ EN SU PRIMER INTENTO ENVIA UNA SOLICITUD A UN SERVIDOR ESPECIAL DE ASOCIACIONES (**SERVIDOR DE NOMBRES**):
 - LE SOLICITA EL N° DE LA MAQUINA DONDE EN ESE MOMENTO SE LOCALIZA EL SERVIDOR.
 - Φ CONOCIENDO LA DIRECCION DEL SERVIDOR, SE LE ENVIA LA SOLICITUD DEL SERVICIO REQUERIDO.
- v OTRO METODO UTILIZA HARDWARE ESPECIAL:
 - u LOS PROCESOS ELIGEN SU DIRECCION EN FORMA ALEATORIA.
 - u LOS CHIPS DE INTERFAZ DE LA RED SE DISEÑAN DE MODO QUE PERMITAN A LOS PROCESOS ALMACENAR DIRECCIONES DE PROCESOS EN ELLOS.
 - u LOS PAQUETES TRANSMITIDOS UTILIZAN DIRECCIONES DE PROCESOS EN VEZ DE DIRECCIONES DE MAQUINAS.
 - u AL RECIBIR CADA PAQUETE EL CHIP DE INTERFAZ DE LA RED DEBE EXAMINARLO PARA DETERMINAR SI EL PROCESO DESTINO SE ENCUENTRA EN ESA MAQUINA:
 - Φ LO ACEPTA EN CASO AFIRMATIVO.
 - Φ NO LO ACEPTA EN CASO NEGATIVO.

PRIMITIVAS DE BLOQUEO VS. NO BLOQUEO EN C - S

- v LAS PRIMITIVAS DE TRANSFERENCIA DE MENSAJES CONSIDERADAS ANTERIORMENTE SE DENOMINAN **PRIMITIVAS DE BLOQUEO O PRIMITIVAS SINCRONAS**:
 - υ EL PROCESO EMISOR SE SUSPENDE (SE BLOQUEA) MIENTRAS SE ENVIA EL MENSAJE.
 - υ EL PROCESO RECEPTOR SE SUSPENDE MIENTRAS SE RECIBE EL MENSAJE.
- v UNA ALTERNATIVA SON LAS **PRIMITIVAS SIN BLOQUEO O PRIMITIVAS ASINCRONAS**:
 - υ EL PROCESO EMISOR:
 - Φ NO SE SUSPENDE MIENTRAS SE ENVIA EL MENSAJE.
 - Φ SI PUEDE CONTINUAR SU COMPUTO PARALELAMENTE CON LA TRANSMISION DEL MENSAJE.
 - Φ NO PUEDE MODIFICAR EL BUFFER DE MENSAJES HASTA QUE SE ENVIE EL MENSAJE.
 - Φ NO TIENE CONTROL SOBRE LA TERMINACION DE LA TRANSMISION:
 - NO SABE CUANDO SERA SEGURO REUTILIZAR EL BUFFER.
- v UNA SOLUCION ES:
 - υ QUE EL NUCLEO COPIE EL MENSAJE A UN BUFFER INTERNO DEL NUCLEO.
 - υ QUE ENTONCES EL NUCLEO PERMITA AL PROCESO CONTINUAR Y REUTILIZAR EL BUFFER.
- v LA DESVENTAJA DE LA SOLUCION ES QUE CADA MENSAJE DE SALIDA DEBE SER COPIADO DESDE EL ESPACIO DEL USUARIO AL ESPACIO DEL NUCLEO.
- v OTRA SOLUCION ES:
 - υ INTERRUMPIR AL EMISOR CUANDO SE ENVIE EL MENSAJE.
 - υ INFORMARLE QUE EL BUFFER NUEVAMENTE ESTA DISPONIBLE.
- v LA DESVENTAJA RADICA EN LA DIFICULTAD DE LA PROGRAMACION BASADA EN INTERRUPCIONES A NIVEL USUARIO.
- v GENERALMENTE SE CONSIDERA QUE LAS DESVENTAJAS DE LAS PRIMITIVAS ASINCRONAS NO COMPENSAN LAS VENTAJAS DEL MAXIMO PARALELISMO QUE PERMITEN LOGRAR.

PRIMITIVAS DE BLOQUEO VS. NO BLOQUEO EN C - S

- v EL CRITERIO UTILIZADO HA SIDO EL SIGUIENTE:
 - u LA DIFERENCIA ESENCIAL ENTRE UNA PRIMITIVA SINCRONA Y UNA ASINCRONA ES SI EL EMISOR PUEDE VOLVER A UTILIZAR EL BUFFER DE MENSAJES EN FORMA INMEDIATA Y SEGURA DESPUES DE RECUPERAR EL CONTROL.
 - u EL MOMENTO EN QUE EL MENSAJE LLEGA AL RECEPTOR ES IRRELEVANTE.
- v OTRO CRITERIO ESTABLECE LO SIGUIENTE:
 - u UNA PRIMITIVA SINCRONA ES AQUELLA EN QUE EL EMISOR SE BLOQUEA HASTA QUE EL RECEPTOR HA ACEPTADO EL MENSAJE Y LA CONFIRMACION REGRESA AL EMISOR.
 - u TODO LO DEMAS ES ASINCRONO CON ESTE CRITERIO.
- v DESDE EL PUNTO DE VISTA DEL S. O. GENERALMENTE SE CONSIDERA EL PRIMER CRITERIO:
 - u EL INTERES ESTA CENTRADO EN EL MANEJO DE LOS BUFFERS Y EN LA TRANSMISION DE LOS MENSAJES.
- v DESDE EL PUNTO DE VISTA DE LOS LENGUAJES DE PROGRAMACION SE TIENDE A CONSIDERAR EL SEGUNDO CRITERIO:
 - u EL INTERES ESTA CENTRADO EN EL LENGUAJE DE PROGRAMACION Y SUS FACILIDADES DE USO.
- v GENERALMENTE A LAS PRIMITIVAS DE ENVIO SE LAS CONOCE COMO *SEND* Y A LAS DE RECEPCION COMO *RECEIVE*:
 - u AMBAS PUEDEN SER CON BLOQUEO O SIN BLOQUEO.
- v UNA RECEPCION SIN BLOQUEO LE INDICA AL NUCLEO LA LOCALIZACION DEL BUFFER Y REGRESA EL CONTROL:
 - u EL PROBLEMA ES SABER QUIEN HIZO LA LLAMADA CUANDO SE LLEVO A CABO LA OPERACION.
 - u LAS SOLUCIONES PUEDEN SER:
 - ⊕ PROPORCIONAR UNA PRIMITIVA EXPLICITA *WAIT* QUE PERMITA AL RECEPTOR BLOQUEARSE CUANDO LO DESEE.
 - ⊕ PROPORCIONAR UNA PRIMITIVA *TEST* QUE PERMITA VERIFICAR EL ESTADO DEL NUCLEO.

PRIMITIVAS ALMACENADAS EN BUFFER VS. NO ALMACENADAS

- v LAS PRIMITIVAS CONSIDERADAS HASTA AHORA SON ESENCIALMENTE **PRIMITIVAS NO ALMACENADAS**:
 - u SIGNIFICA QUE UNA DIRECCION SE REFIERE A UN PROCESO ESPECIFICO.
 - u UNA LLAMADA *receive (addr, &m)* LE INDICA AL NUCLEO DE LA MAQUINA EN DONDE SE EJECUTA:
 - Φ QUE EL PROCESO QUE HACE LA LLAMADA ESCUCHA A LA DIRECCION *addr*.
 - Φ QUE ESTA PREPARADA PARA RECIBIR EL MENSAJE ENVIADO A ESA DIRECCION.
 - Φ QUE SE DISPONE DE UN UNICO BUFFER DE MENSAJES AL QUE APUNTA *m* PARA CAPTURAR EL MENSAJE QUE LLEGARA.
 - Φ QUE CUANDO EL MENSAJE LLEGUE SERA COPIADO (POR EL NUCLEO RECEPTOR) AL BUFFER:
 - SE ELIMINA ENTONCES EL BLOQUEO DEL PROCESO RECEPTOR.
- v ESTE ESQUEMA FUNCIONA BIEN CUANDO EL SERVIDOR LLAMA A *RECEIVE* ANTES DE QUE EL CLIENTE LLAME A *SEND*.
- v EL PROBLEMA SE PRESENTA CUANDO EL *SEND* SE LLEVA A CABO ANTES QUE EL *RECEIVE*:
 - u EL NUCLEO DEL SERVIDOR:
 - Φ NO SABE CUAL DE SUS PROCESOS UTILIZA LA DIRECCION EN EL MENSAJE RECIEN LLEGADO.
 - Φ NO SABE DONDE COPIAR EL MENSAJE RECIBIDO.
- v UNA SOLUCION CONSISTE EN:
 - u DESCARTAR EL MENSAJE.
 - u DEJAR QUE EL CLIENTE ESPERE.
 - u CONFIAR EN QUE EL SERVIDOR LLAME A *RECEIVE* ANTES DE QUE EL CLIENTE VUELVA A TRANSMITIR:
 - Φ EL CLIENTE PODRIA TENER QUE INTENTAR VARIAS VECES.

PRIMITIVAS ALMACENADAS EN BUFFER VS. NO ALMACENADAS

- v SI DOS O MAS CLIENTES UTILIZAN UN SERVIDOR CON TRANSFERENCIA DE MENSAJES SIN ALMACENAMIENTO EN BUFFERS:
 - u LUEGO DE QUE EL SERVIDOR ACEPTO UN MENSAJE DE UNO DE ELLOS:
 - Φ DEJA DE ESCUCHAR A SU DIRECCION HASTA QUE TERMINA SU TRABAJO.
 - Φ REGRESA AL PRINCIPIO DEL CICLO PARA VOLVER A LLAMAR A *RECEIVE*.
 - u SI REALIZAR EL TRABAJO INSUME CIERTO TIEMPO:
 - Φ LOS DEMAS CLIENTES PODRIAN HACER VARIOS INTENTOS DE ENVIOS SIN EXITO.
- v OTRA SOLUCION CONSISTE EN HACER QUE EL NUCLEO RECEPTOR MANTENGA PENDIENTES LOS MENSAJES POR UN INSTANTE:
 - u PARA PREVENIR QUE UN *RECEIVE* ADECUADO SE REALICE EN UN TIEMPO CORTO.
 - u CUANDO LLEGA UN MENSAJE “NO DESEADO”, SE INICIALIZA EL CRONOMETRO:
 - Φ SI EL TIEMPO EXPIRA ANTES DE QUE OCURRA UN *RECEIVE* APROPIADO, EL MENSAJE SE DESCARTA.
 - u SE REDUCE LA PROBABILIDAD DE QUE UN MENSAJE SE PIERDA.
 - u SE DEBE ALMACENAR Y MANEJAR LOS MENSAJES QUE LLEGAN EN FORMA PREMATURA.
 - u SE NECESITAN LOS BUFFERS Y LA ADMINISTRACION DE LOS MISMOS.
 - u SE PUEDE HACER MEDIANTE UNA NUEVA ESTRUCTURA DE DATOS LLAMADA **BUZON**:
 - Φ UN PROCESO INTERESADO EN RECIBIR MENSAJES:
 - LE INDICA AL NUCLEO QUE CREE UN BUZON PARA EL.
 - ESPECIFICA UNA DIRECCION EN LA CUAL BUSCAR LOS PAQUETES DE LA RED.
 - Φ TODOS LOS MENSAJES QUE LLEGUEN EN ESA DIRECCION SE COLOCAN EN EL BUZON.

PRIMITIVAS ALMACENADAS EN BUFFER VS. NO ALMACENADAS

- v LA LLAMADA A *RECEIVE* ELIMINA UN MENSAJE DEL BUZON O SE BLOQUEA (SI SE UTILIZAN PRIMITIVAS CON BLOQUEO) SI NO HAY UN MENSAJE PRESENTE.
- v ESTA TECNICA SE DENOMINA **PRIMITIVA CON ALMACENAMIENTO EN BUFFERS**.
- v LOS BUZONES TIENEN EL PROBLEMA DE QUE SON FINITOS Y PUEDEN OCUPARSE EN SU TOTALIDAD:
 - u CUANDO LLEGA UN MENSAJE A UN BUZON LLENO, EL NUCLEO DEBE ELEGIR ENTRE:
 - Φ MANTENER EL MENSAJE PENDIENTE POR UN MOMENTO ESPERANDO QUE ALGUN MENSAJE SEA RETIRADO DEL BUZON A TIEMPO.
 - Φ DESCARTAR EL MENSAJE.
 - u ESTA ES LA MISMA SITUACION QUE SE TIENE CUANDO SE TRABAJA SIN ALMACENAMIENTO EN BUFFERS:
 - Φ CON BUFFERS SE REDUCE LA PROBABILIDAD DE PROBLEMAS, PERO LOS PROBLEMAS NO SE ELIMINAN NI CAMBIA SU NATURALEZA.
- v OTRA SOLUCION UTILIZADA ES NO DEJAR QUE UN PROCESO ENVIE UN MENSAJE SI NO EXISTE ESPACIO PARA SU ALMACENAMIENTO EN EL DESTINO:
 - u EL EMISOR DEBE BLOQUEARSE HASTA QUE OBTENGA DE REGRESO UN RECONOCIMIENTO:
 - Φ DEBE INDICAR QUE EL MENSAJE HA SIDO RECIBIDO.
 - u SI EL BUZON ESTA LLENO, EL EMISOR PUEDE HACER UN RESPALDO Y SUSPENDERSE DE MANERA RETROACTIVA:
 - Φ LA SUSPENSION DEBE OPERAR COMO SI FUERA JUSTO ANTES DE QUE INTENTARA ENVIAR EL MENSAJE.
 - Φ CUANDO HAYA ESPACIO LIBRE EN EL BUZON, SE HARA QUE EL EMISOR INTENTE NUEVAMENTE.

PRIMITIVAS CONFIABLES VS. NO CONFIABLES

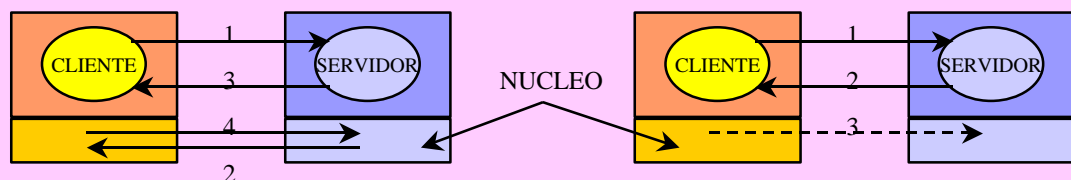
- v HASTA ACA SE HA SUPUESTO QUE LOS MENSAJES ENVIADOS SIEMPRE SERAN RECIBIDOS:
 - u EN LA REALIDAD, LOS MENSAJES SE PUEDEN PERDER POR DIVERSAS CAUSAS.
- v CUANDO UN CLIENTE ENVIA UN MENSAJE SE LE PUEDE SUSPENDER HASTA QUE EL MENSAJE HA SIDO ENVIADO:
 - u CUANDO CONTINUA, NO HAY GARANTIA DE QUE EL MENSAJE HA SIDO ENTREGADO:
 - Φ EL MENSAJE PODRIA HABERSE PERDIDO.
- v UN ENFOQUE DE ESTE PROBLEMA CONSISTE EN VOLVER A DEFINIR LA SEMANTICA DE *SEND* PARA HACERLO NO CONFIABLE:
 - u EL SISTEMA NO GARANTIZA LA ENTREGA DE LOS MENSAJES.
 - u LA IMPLANTACION DE UNA COMUNICACION CONFIABLE SE DEJA EN MANOS DE LOS USUARIOS.
- v OTRO METODO EXIGE QUE EL NUCLEO DE LA MAQUINA RECEPTORA ENVIE UN RECONOCIMIENTO AL NUCLEO DE LA MAQUINA EMISORA:
 - u SOLO CUANDO RECIBA EL RECONOCIMIENTO, EL NUCLEO EMISOR LIBERARA AL PROCESO USUARIO (CLIENTE).
 - u LA SOLICITUD DE UN CLIENTE A UN SERVIDOR ES RECONOCIDA POR EL NUCLEO DEL SERVIDOR.
 - u LA RESPUESTA DEL SERVIDOR DE REGRESO AL CLIENTE ES RECONOCIDA POR EL NUCLEO DEL CLIENTE.
 - u UNA SOLICITUD DE RESPUESTA CONSTA DE CUATRO MENSAJES.
- v OTRA SOLUCION APROVECHA EL HECHO DE QUE LA COMUNICACION CLIENTE - SERVIDOR SE ESTRUCTURA:
 - u COMO UNA SOLICITUD DEL CLIENTE AL SERVIDOR.
 - u SEGUIDA DE UNA RESPUESTA DEL SERVIDOR AL CLIENTE.
 - u EL CLIENTE SE BLOQUEA DESPUES DE ENVIAR UN MENSAJE.
 - u EL NUCLEO DEL SERVIDOR NO ENVIA DE REGRESO UN RECONOCIMIENTO SINO QUE LA MISMA RESPUESTA FUNCIONA COMO TAL.
 - u EL EMISOR PERMANECE BLOQUEADO HASTA QUE REGRESA LA RESPUESTA.
 - u SI LA RESPUESTA NO LLEGA EN CIERTO TIEMPO, EL NUCLEO EMISOR PUEDE VOLVER A ENVIAR LA SOLICITUD:
 - Φ SE PROTEGE CONTRA LA POSIBILIDAD DE UNA PERDIDA DEL MENSAJE.

PRIMITIVAS CONFIABLES VS. NO CONFIABLES

- v EN EL ESQUEMA ANTERIOR LA RESPUESTA FUNCIONA COMO UN RECONOCIMIENTO A LA SOLICITUD:
 - u NO EXISTE UN RECONOCIMIENTO POR LA RESPUESTA:
 - Φ LA SERIEDAD DE ESTA OMISION DEPENDE DE LA NATURALEZA DE LA SOLICITUD.
 - u EN ALGUNOS CASOS SE UTILIZA UN RECONOCIMIENTO DEL NUCLEO DEL CLIENTE AL NUCLEO DEL SERVIDOR:
 - Φ HASTA NO RECIBIR ESTE PAQUETE, EL *SEND* DEL SERVIDOR NO TERMINA:
 - EL SERVIDOR PERMANECE BLOQUEADO (SI SE UTILIZAN PRIMITIVAS CON BLOQUEO).
- v OTRA POSIBILIDAD ES LA SIGUIENTE:
 - u AL LLEGAR UNA SOLICITUD AL NUCLEO DEL SERVIDOR, SE INICIA UN CRONOMETRO.
 - u SI EL SERVIDOR ENVIA LA RESPUESTA ANTES DE QUE TERMINE EL CRONOMETRO, ESTA FUNCIONA COMO EL RECONOCIMIENTO.
 - u SI EXPIRA EL TIEMPO DEL CRONOMETRO, SE ENVIA UN RECONOCIMIENTO SEPARADO.

MENSAJES RECONOCIDOS EN FORMA INDIVIDUAL

LA RESPUESTA SE UTILIZA COMO RECONOCIMIENTO DE LA SOLICITUD



1. SOLICITUD (DEL CLIENTE AL SERVIDOR).
2. RECONOCIMIENTO (DE NUCLEO A NUCLEO).
3. RESPUESTA (DEL SERVIDOR AL CLIENTE).
4. RECONOCIMIENTO (DE NUCLEO A NUCLEO).

1. SOLICITUD (DEL CLIENTE AL SERVIDOR).
2. RESPUESTA (DEL SERVIDOR AL CLIENTE).
3. RECONOCIMIENTO (DE NUCLEO A NUCLEO) - NO SIEMPRE EXISTE, POR ELLO APARECE COMO LINEA DISCONTINUA.

IMPLANTACION DEL MODELO C - S

- v LAS PRINCIPALES OPCIONES DE DISEÑO ANALIZADAS SE RESUMEN EN:
 - u DIRECCIONAMIENTO:
 - Φ NUMERO DE MAQUINA.
 - Φ DIRECCIONES RALAS DE PROCESOS.
 - Φ BUSQUEDA DE NOMBRES EN ASCII POR MEDIO DEL SERVIDOR.
 - u BLOQUEO:
 - Φ PRIMITIVAS CON BLOQUEO.
 - Φ SIN BLOQUEO, CON COPIA AL NUCLEO.
 - Φ SIN BLOQUEO, CON INTERRUPCIONES.
 - u ALMACENAMIENTO EN BUFFERS:
 - Φ NO USAR EL ALMACENAMIENTO EN BUFFERS, DESCARTAR LOS MENSAJES INESPERADOS.
 - Φ SIN ALMACENAMIENTO EN BUFFERS, MANTENIMIENTO TEMPORAL DE LOS MENSAJES INESPERADOS.
 - Φ BUZONES.
 - u CONFIABILIDAD:
 - Φ NO CONFIABLE.
 - Φ SOLICITUD - RECONOCIMIENTO - RESPUESTA - RECONOCIMIENTO.
 - Φ SOLICITUD - RESPUESTA - RECONOCIMIENTO.
- v EXISTEN $3^4 = 81$ COMBINACIONES:
 - u NO TODAS SON IGUAL DE BUENAS.
- v EN EL CASO DE MENSAJES COMPUESTOS POR VARIOS PAQUETES, EL RECONOCIMIENTO PUEDE SER:
 - u POR PAQUETE INDIVIDUAL:
 - Φ ANTE LA PERDIDA DE UN PAQUETE, SOLO RETRANSMITE ESE PAQUETE.
 - Φ REQUIERE MAS PAQUETES EN LA RED.
 - u POR MENSAJE COMPLETO:
 - Φ LA RECUPERACION ES COMPLEJA ANTE LA PERDIDA DE UN PAQUETE.
 - Φ REQUIERE MENOS PAQUETES EN LA RED.

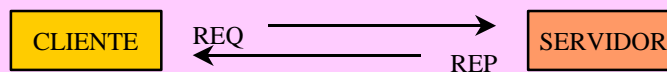
IMPLANTACION DEL MODELO C - S

- v OTRO ASPECTO INTERESANTE DE LA IMPLEMENTACION ES EL PROTOCOLO SUBYACENTE UTILIZADO EN LA COMUNICACION C - S.
- v LOS PRINCIPALES TIPOS DE PAQUETES SON LOS SIGUIENTES:
 - u REQ:
 - Φ SOLICITUD.
 - Φ DE CLIENTE A SERVIDOR.
 - Φ EL CLIENTE DESEA SERVICIO.
 - u REP:
 - Φ RESPUESTA.
 - Φ DE SERVIDOR A CLIENTE.
 - Φ RESPUESTA DEL SERVIDOR AL CLIENTE.
 - u ACK:
 - Φ RECONOCIMIENTO.
 - Φ DE CUALQUIERA DE ELLOS A ALGUN OTRO.
 - Φ EL PAQUETE ANTERIOR QUE HA LLEGADO.
 - u AYA:
 - Φ ¿ESTAS VIVO?.
 - Φ DE CLIENTE A SERVIDOR.
 - Φ VERIFICA SI EL SERVIDOR SE HA DESCOMPUESTO.
 - u IAA:
 - Φ ESTOY VIVO.
 - Φ DE SERVIDOR A CLIENTE.
 - Φ EL SERVIDOR NO SE HA DESCOMPUESTO.
 - u TA:
 - Φ INTENTA DE NUEVO.
 - Φ DE SERVIDOR A CLIENTES.
 - Φ EL SERVIDOR NO TIENE ESPACIO.
 - u AU:
 - Φ DIRECCION DESCONOCIDA.
 - Φ DE SERVIDOR A CLIENTE.
 - Φ NINGUN PROCESO UTILIZA ESTA DIRECCION.

IMPLANTACION DEL MODELO C - S

EJEMPLOS DE INTERCAMBIO DE PAQUETES PARA LA COMUNICACION CLIENTE - SERVIDOR

SOLICITUD / RESPUESTA DIRECTA, SIN RECONOCIMIENTOS



RECONOCIMIENTO DE CADA MENSAJE INDIVIDUAL



LA RESPUESTA ACTUA COMO RECONOCIMIENTO



VERIFICACION DE UN CLIENTE RESPECTO DE SI EL SERVIDOR ESTA ACTIVO



LLAMADA A UN PROCEDIMIENTO REMOTO (RPC)

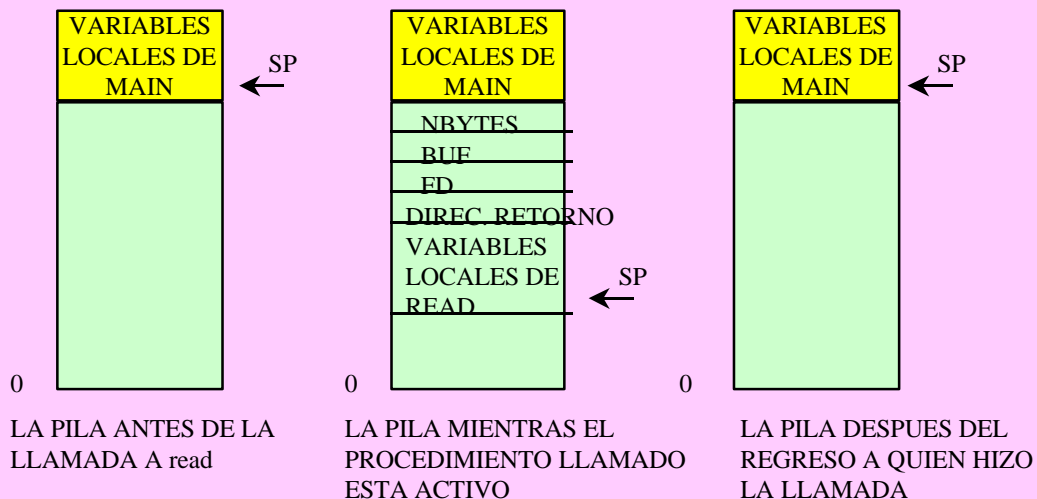
- v EL MODELO CLIENTE - SERVIDOR ES UNA FORMA CONVENIENTE DE ESTRUCTURAR UN S. O. DISTRIBUIDO, PERO POSEE UNA FALENCIA:
 - u EL PARADIGMA ESENCIAL EN TORNO AL QUE SE CONSTRUYE LA COMUNICACION ES LA ENTRADA / SALIDA.
 - u LOS PROCEDIMIENTOS *SEND* / *RECEIVE* ESTAN RESERVADOS PARA LA REALIZACION DE E / S.
- v UNA OPCION DISTINTA FUE PLANTEADA POR BIRREL Y NELSON:
 - u PERMITIR A LOS PROGRAMAS QUE LLAMASEN A PROCEDIMIENTOS LOCALIZADOS EN OTRAS MAQUINAS.
 - u CUANDO UN PROCESO EN LA MAQUINA A LLAMA A UN PROCEDIMIENTO EN LA MAQUINA B:
 - Φ EL PROCESO QUE REALIZA LA LLAMADA SE SUSPENDE.
 - Φ LA EJECUCION DEL PROCEDIMIENTO SE REALIZA EN B.
 - u LA INFORMACION SE PUEDE TRANSPORTAR DE UN LADO AL OTRO MEDIANTE LOS PARAMETROS Y PUEDE REGRESAR EN EL RESULTADO DEL PROCEDIMIENTO.
 - u EL PROGRAMADOR NO SE PREOCUPA DE UNA TRANSFERENCIA DE MENSAJES O DE LA E / S.
 - u A ESTE METODO SE LO DENOMINA **LLAMADA A PROCEDIMIENTO REMOTO O RPC**.
 - u EL PROCEDIMIENTO QUE HACE LA LLAMADA Y EL QUE LA RECIBE SE EJECUTAN EN MAQUINAS DIFERENTES:
 - Φ UTILIZAN ESPACIOS DE DIRECCIONES DISTINTOS.

OPERACION BASICA DE RPC

- v UNA LLAMADA CONVENCIONAL A UN PROCEDIMIENTO, ES DECIR EN UNA SOLA MAQUINA, FUNCIONA DE LA SIGUIENTE MANERA:
 - u SEA `count = read (fd, buf, nbytes);` DONDE:
 - ⊕ `fd` ES UN ENTERO; `buf` ES UN ARREGLO DE CARACTERES; `nbytes` ES OTRO ENTERO.
 - u EL PROGRAMA LLAMADOR COLOCA LOS PARAMETROS EN LA PILA.
 - u EL PROCEDIMIENTO LLAMADO DESDE EL PROGRAMA LLAMADOR SE CARGA EN LA MEMORIA.
 - u DESPUES DE QUE `read` TERMINA SU EJECUCION:
 - ⊕ COLOCA EL VALOR DE REGRESO EN UN REGISTRO.
 - ⊕ ELIMINA LA DIRECCION DE REGRESO.
 - ⊕ TRANSFIERE DE NUEVO EL CONTROL A QUIEN HIZO LA LLAMADA.
 - ⊕ QUIEN HIZO LA LLAMADA ELIMINA LOS PARAMETROS DE LA PILA Y REGRESA A SU ESTADO ORIGINAL.
- v LOS PARAMETROS PUEDEN **LLAMARSE POR VALOR O POR REFERENCIA.**
- v UN **PARAMETRO POR VALOR:**
 - u SE COPIA A LA PILA.
 - u PARA EL PROCEDIMIENTO QUE RECIBE LA LLAMADA ES SOLO UNA VARIABLE LOCAL YA INICIALIZADA.
 - u EL PROCEDIMIENTO PODRIA MODIFICARLA, SIN QUE ESTO AFECTE EL VALOR DE LA VARIABLE ORIGINAL EN EL PROCEDIMIENTO QUE HIZO LA LLAMADA.
- v UN **PARAMETRO POR REFERENCIA:**
 - u ES UN APUNTADOR A UNA VARIABLE (ES DECIR, LA DIRECCION DE LA VARIABLE), NO EL VALOR DE LA VARIABLE.
 - u EN EL EJ. ANTERIOR, VALIDO PARA "C", EL SEGUNDO PARAMETRO ES UN PARAMETRO POR REFERENCIA Y ES UN ARREGLO.
 - u SI EL PROCEDIMIENTO QUE RECIBE LA LLAMADA UTILIZA ESTE PARAMETRO POR REFERENCIA PARA ALMACENAR ALGO EN EL ARREGLO:
 - ⊕ MODIFICA EL ARREGLO EN EL PROCEDIMIENTO QUE HIZO LA LLAMADA

OPERACION BASICA DE RPC

LLAMADA A UN PROCEDIMIENTO LOCAL



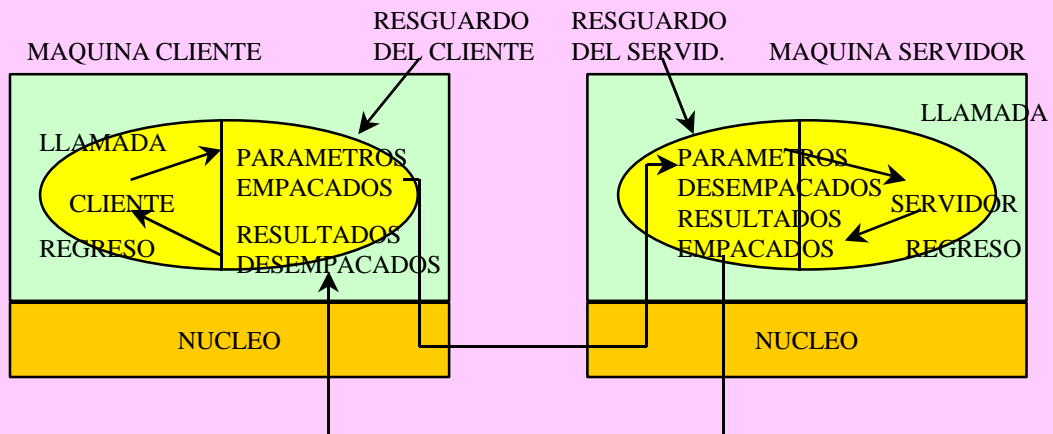
- v OTRO MECANISMO PARA EL PASO DE PARAMETROS ES LA **LLAMADA POR COPIAR / RESTAURAR**:
 - u QUIEN RECIBE LA LLAMADA COPIA LA VARIABLE EN LA PILA, COMO EN LA LLAMADA POR VALOR.
 - u LA COPIA DE NUEVO DESPUES DE LA LLAMADA, ESCRIBIENDO SOBRE EL VALOR ORIGINAL.
- v LA DECISION DE CUAL MECANISMO UTILIZAR PARA EL PASO DE PARAMETROS:
 - u LA TOMAN LOS DISEÑADORES DEL SISTEMA Y ES UNA PROPIEDAD FIJA DEL LENGUAJE.
- v LA IDEA ES QUE UNA LLAMADA A UN PROCEDIMIENTO REMOTO (RPC) SE PAREZCA LO MAS POSIBLE A UNA LLAMADA LOCAL:
 - u LA RPC DEBE SER TRANSPARENTE.
 - u EL PROCEDIMIENTO QUE HACE LA LLAMADA NO DEBE SER CONSCIENTE DE QUE EL PROCEDIMIENTO LLAMADO SE EJECUTA EN UNA MAQUINA DISTINTA, O VICEVERSA.
 - u SI read ES UN PROCEDIMIENTO REMOTO (EJ.: SE EJECUTA EN LA MAQUINA DEL SERVIDOR DE ARCHIVOS) SE COLOCA EN LA BIBLIOTECA UNA VERSION DISTINTA DE read LLAMADA **STUB DEL CLIENTE**:
 - ⊕ NO COLOCA LOS PARAMETROS EN REGISTROS Y LE PIDE AL NUCLEO QUE LE PROPORCIONE DATOS.
 - ⊕ COLOCA LOS PARAMETROS EN UN MENSAJE Y LE PIDE LA NUCLEO QUE ENVIE EL MENSAJE AL SERVIDOR.

OPERACION BASICA DE RPC

- ❖ DESPUES DE LA LLAMADA A send, EL STUB DEL CLIENTE LLAMA A receive Y SE BLOQUEA HASTA QUE REGRESE LA RESPUESTA.
- v CUANDO EL MENSAJE LLEGA AL SERVIDOR:
 - u EL NUCLEO LO TRANSFIERE A UN **STUB DEL SERVIDOR**.
 - u GENERALMENTE EL STUB DEL SERVIDOR YA HABRA LLAMADO A receive Y ESTARA BLOQUEADO ESPERANDO QUE LE LLEGUEN MENSAJES.
 - u EL STUB DEL SERVIDOR:
 - ❖ “DESEMPACA” LOS PARAMETROS DEL MENSAJE.
 - ❖ LLAMA AL PROCEDIMIENTO DEL SERVIDOR DE LA MANERA CONVENCIONAL.
- v PARA EL SERVIDOR ES COMO SI TUVIERA UNA LLAMADA DIRECTA DEL CLIENTE:
 - u LLEVA A CABO EL TRABAJO Y REGRESA EL RESULTADO A QUIEN HIZO LA LLAMADA, DE LA FORMA USUAL.
- v EL STUB DEL SERVIDOR RECUPERA EL CONTROL LUEGO DE LA LLAMADA Y:
 - u EMPACA EL RESULTADO EN UN MENSAJE.
 - u LLAMA A send PARA REGRESARLO AL CLIENTE.
 - u LLAMA A receive Y ESPERA EL SIGUIENTE MENSAJE.
- v CUANDO EL MENSAJE REGRESA A LA MAQUINA CLIENTE:
 - u EL NUCLEO VE QUE ESTA DIRIGIDO AL PROCESO CLIENTE.
 - u EL MENSAJE SE COPIA AL BUFFER EN ESPERA.
 - u EL PROCESO CLIENTE ELIMINA SU BLOQUEO.
 - u EL STUB DEL CLIENTE EXAMINA EL MENSAJE, DESEMPACA EL RESULTADO, LO COPIA A QUIEN HIZO LA LLAMADA Y REGRESA DE LA MANERA USUAL.
- v CUANDO EL PROCESO QUE HIZO LA LLAMADA OBTIENE EL CONTROL LUEGO DE LA LLAMADA A read:
 - u DISPONE DE LOS DATOS.
 - u IGNORA QUE EL TRABAJO SE REALIZO DE MANERA REMOTA.
 - u *HA TENIDO ACCESO A SERVICIOS REMOTOS MEDIANTE LLAMADAS COMUNES A PROCEDIMIENTOS LOCALES.*

OPERACION BASICA DE RPC

LLAMADAS Y MENSAJES EN UNA RPC. CADA ELIPSE REPRESENTA UN SOLO PROCESO, QUE INCLUYE EL RESGUARDO.



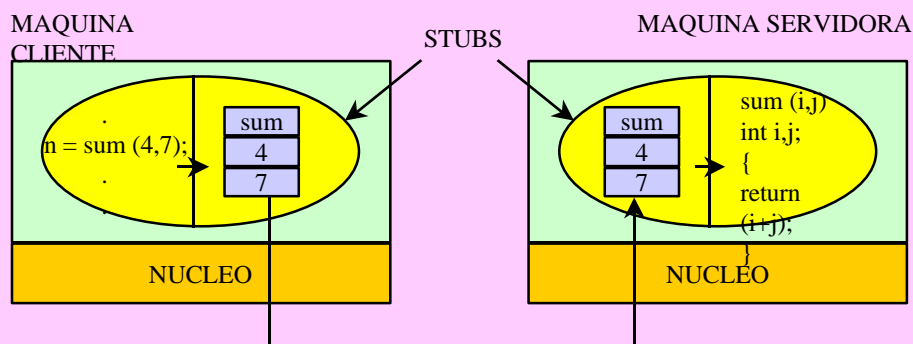
TRANSPORTE DE UN MENSAJE SOBRE LA RED

- v RESUMIENDO, SE PUEDEN INDICAR LOS SIGUIENTES PASOS COMO UNA LLAMADA A UN PROCEDIMIENTO REMOTO:
 - v EL PROCEDIMIENTO CLIENTE LLAMA AL STUB DEL CLIENTE DE LA MANERA USUAL.
 - v EL STUB DEL CLIENTE CONSTRUYE UN MENSAJE Y HACE UN SEÑALAMIENTO AL NUCLEO.
 - v EL NUCLEO ENVIA EL MENSAJE AL NUCLEO REMOTO.
 - v EL NUCLEO REMOTO PROPORCIONA EL MENSAJE AL STUB DEL SERVIDOR.
 - v EL STUB DEL SERVIDOR DESEMPACA LOS PARAMETROS Y LLAMA AL SERVIDOR.
 - v EL SERVIDOR REALIZA EL TRABAJO Y REGRESA EL RESULTADO AL STUB.
 - v EL STUB DEL SERVIDOR EMPACA EL RESULTADO EN UN MENSAJE Y HACE UN SEÑALAMIENTO AL NUCLEO.
 - v EL NUCLEO REMOTO ENVIA EL MENSAJE AL NUCLEO DEL CLIENTE.
 - v EL NUCLEO DEL CLIENTE DA EL MENSAJE AL STUB DEL CLIENTE.
 - v EL STUB DESEMPACA EL RESULTADO Y REGRESA AL CLIENTE.
- v SE HA CONVERTIDO LA LLAMADA LOCAL DEL PROCEDIMIENTO CLIENTE AL STUB DEL CLIENTE, EN UNA LLAMADA LOCAL AL PROCEDIMIENTO SERVIDOR.

TRANSFERENCIA DE PARAMETROS EN RPC

- v EL EMPACAMIENTO DE PARAMETROS EN UN MENSAJE SE LLAMA **ORDENAMIENTO DE PARAMETROS**.
- v EL MENSAJE TAMBIEN CONTIENE EL NOMBRE O N° DEL PROCEDIMIENTO POR LLAMAR:
 - EL SERVIDOR PODRIA SOPORTAR VARIAS LLAMADAS Y SE EL TIENE QUE INDICAR CUAL DE ELLAS SE NECESITA.
- v CUANDO EL MENSAJE LLEGA AL SERVIDOR:
 - EL RESGUARDO (STUB) LO EXAMINA PARA VER CUAL PROCEDIMIENTO NECESITA.
 - LLEVA A CABO LA LLAMADA APROPIADA.
- v LA LLAMADA REAL DEL RESGUARDO AL SERVIDOR ES SIMILAR A LA LLAMADA ORIGINAL DEL CLIENTE, EXCEPTO EN QUE:
 - LOS PARAMETROS SON VARIABLES INICIALIZADAS A PARTIR DEL MENSAJE RECIBIDO, EN VEZ DE SER CONSTANTES.
- v LOS ELEMENTOS DEL MENSAJE CORRESPONDEN A:
 - IDENTIFICADOR DEL PROCEDIMIENTO.
 - PARAMETROS.
- v UN MENSAJE QUE CORRESPONDA A UN PROCEDIMIENTO REMOTO CON n PARAMETROS TENDRA $n + 1$ CAMPOS:
 - UNO PARA IDENTIFICAR AL PROCEDIMIENTO.
 - UNO PARA C / U DE LOS n PARAMETROS.

CALCULO REMOTO DE `sum (4,7)`



TRANSFERENCIA DE PARAMETROS EN RPC

- v UN ASPECTO IMPORTANTE ES DETERMINAR COMO SE DEBE REPRESENTAR LA INFORMACION EN LOS MENSAJES:
 - u UNA FORMA ES:
 - Φ DISEÑAR UN ESTANDAR DE RED O **FORMA CANONICA** PARA LOS ENTEROS, CARACTERES, BOOLEANOS, N° DE PUNTO FLOTANTE, ETC.
 - Φ PEDIR A TODOS LOS EMISORES QUE CONVIERTAN SUS REPRESENTACIONES INTERNAS A ESTA FORMA DURANTE EL ORDENAMIENTO.
- v ES IMPORTANTE SABER LA ORGANIZACION DEL MENSAJE Y LA IDENTIDAD DEL CLIENTE.
- v TAMBIEN ES IMPORTANTE DETERMINAR DE DONDE PROVIENEN LOS PROCEDIMIENTOS RESGUARDO:
 - u EN MUCHOS SISTEMAS SE GENERAN AUTOMATICAMENTE.
 - u DADA UNA ESPECIFICACION DEL PROCEDIMIENTO SERVIDOR Y LAS REGLAS DE CODIFICACION:
 - Φ EL FORMATO DEL MENSAJE QUEDA DETERMINADO DE MANERA UNICA.
- v ES POSIBLE TENER UN COMPILADOR QUE:
 - u LEA LAS ESPECIFICACIONES DEL SERVIDOR.
 - u GENERE UN RESGUARDO DEL CLIENTE QUE EMPAQUE SUS PARAMETROS EN EL FORMATO ESTANDAR DE LOS MENSAJES.
 - u GENERE UN RESGUARDO DEL SERVIDOR QUE LOS DESEMPAQUE.
 - u LLAME AL SERVIDOR.
- v UN ASPECTO TAMBIEN MUY IMPORTANTE ES COMO SE TRANSFIEREN LOS APUNTADES:
 - u UN APUNTADES SOLO TIENE SENTIDO DENTRO DEL ESPACIO DE DIRECCIONES DEL PROCESO EN EL QUE SE UTILIZA.
- v UNA SOLUCION ES COPIAR EN EL MENSAJE LA ESTRUCTURA PARA LA CUAL SE UTILIZA EL APUNTADES Y ENVIARLO AL SERVIDOR:
 - u LOS CAMBIOS QUE REALICE EL SERVIDOR MEDIANTE EL APUNTADES AFECTARAN DIRECTAMENTE AL BUFFER DE MENSAJES EN EL RESGUARDO DEL SERVIDOR.

CONEXION DINAMICA (DYNAMIC BINDING) EN RPC

- v UN TEMA FUNDAMENTAL ES LA FORMA EN QUE EL CLIENTE LOCALIZA AL SERVIDOR.
- v UN METODO CONSISTE EN INTEGRAR DENTRO DEL CODIGO DEL CLIENTE LA DIRECCION (EN LA RED) DEL SERVIDOR:
 - u EL PROBLEMA ES QUE RESULTA DEMASIADO RIGIDO.
 - u SI EL SERVIDOR SE DESPLAZA, SI SE DUPLICA O SI CAMBIA LA INTERFAZ:
 - Φ HABRIA QUE LOCALIZAR Y VOLVER A COMPILAR LOS NUMEROSOS PROGRAMAS.
- v UNA SOLUCION ES LA **CONEXION DINAMICA** PARA QUE CONCUERDEN LOS CLIENTES Y LOS SERVIDORES.
- v EL PUNTO DE INICIO DE LA CONEXION DINAMICA ES LA ESPECIFICACION FORMAL DEL SERVIDOR:
 - u INDICA EL NOMBRE DEL SERVIDOR, EL N° DE VERSION Y UNA LISTA DE LOS PROCEDIMIENTOS QUE PROPORCIONA.
- v SE TIENEN LOS TIPOS DE PARAMETROS PARA CADA PROCEDIMIENTO:
 - u CADA PARAMETRO QUEDA DETERMINADO COMO PARAMETRO in, out O in-out.
- v LA DIRECCION ES RELATIVA AL SERVIDOR.
- v EL PRINCIPAL USO DE LA ESPECIFICACION FORMAL ES COMO ENTRADA DEL GENERADOR DE RESGUARDOS:
 - u PRODUCE EL RESGUARDO DEL CLIENTE Y EL DEL SERVIDOR.
 - u AMBOS RESGUARDOS SE COLOCAN EN LAS BIBLIOTECAS RESPECTIVAS.
- v CUANDO UN PROGRAMA (CLIENTE) LLAMA A CUALQUIERA DE LOS PROCEDIMIENTOS DEFINIDOS MEDIANTE ESA ESPECIFICACION:
 - u EL CORRESPONDIENTE PROCEDIMIENTO RESGUARDO DEL CLIENTE SE LIGA CON SU BINARIO.
- v SI SE COMPILA UN PROGRAMA SERVIDOR:
 - u LOS RESGUARDOS DEL SERVIDOR SE LE LIGAN TAMBIEN.

CONEXION DINAMICA (DYNAMIC BINDING) EN RPC

- v CUANDO EL SERVIDOR INICIA SU EJECUCION:
 - u UNA LLAMADA TIPO `initialize` QUE SE ENCUENTRA FUERA DEL CICLO PRINCIPAL **EXPORTA** LA INTERFAZ DEL SERVIDOR:
 - ⊕ EL SERVIDOR ENVIA UN MENSAJE A UN PROGRAMA **CONECTOR** PARA DARLE A CONOCER SU EXISTENCIA.
 - ⊕ ESTO ES EL **REGISTRO** DEL SERVIDOR (**REGISTERING THE SERVER**).
 - ⊕ EL SERVIDOR PROPORCIONA AL CONECTOR SU NOMBRE, N° DE VERSION, UN UNICO IDENTIFICADOR.
 - ⊕ EL IDENTIFICADOR GENERALMENTE, TIENE UNA LONGITUD DE 32 BITS Y UN **ASA (HANDLE)** QUE SE UTILIZA PARA LOCALIZARLO.
 - ⊕ EL ASA (HANDLE) DEPENDE DEL SISTEMA Y PUEDE SER:
 - UNA DIRECCION ETHERNET, IP, X.500.
 - UN IDENTIFICADOR RALO DE PROCESOS, ETC.
 - ⊕ EL ASA TAMBIEN PUEDE PROPORCIONAR INFORMACION RELATIVA A LA AUTENTICACION.
- v UN SERVIDOR PUEDE CANCELAR SU REGISTRO CON EL CONECTOR SI YA NO ESTA PREPARADO PARA PRESTAR ALGUN SERVICIO.
- v EL CLIENTE LOCALIZA AL SERVIDOR DE LA SIGUIENTE MANERA:
 - u CUANDO EL CLIENTE LLAMA A ALGUNO DE LOS PROCEDIMIENTOS REMOTOS POR PRIMERA VEZ:
 - ⊕ EL RESGUARDO DEL CLIENTE:
 - VE QUE AUN NO ESTA CONECTADO A UN SERVIDOR.
 - ENVIA UN MENSAJE AL CONECTOR SOLICITANDO LA **IMPORTACION** DE CIERTA VERSION DE CIERTA INTERFAZ.
 - ⊕ EL CONECTOR VERIFICA SI UNO O MAS SERVIDORES YA HAN EXPORTADO UNA INTERFAZ CON ESE NOMBRE Y VERSION.
 - ⊕ SI NINGUNO DE LOS SERVIDORES EN EJECUCION EN ESE MOMENTO SOPORTA ESA INTERFAZ, LA LLAMADA FRACASA.
 - ⊕ SI EXISTE UN SERVIDOR ADECUADO, EL CONECTOR PROPORCIONA UN ASA E IDENTIFICADOR UNICO AL RESGUARDO DEL CLIENTE:
 - UTILIZA EL ASA COMO LA DIRECCION A LA CUAL ENVIAR EL MENSAJE SOLICITADO.
- v ES UN ESQUEMA MUY FLEXIBLE PERO EL CONECTOR PUEDE SER UN CUELLO DE BOTELLA CON ALTAS CARGAS DE TRABAJO.

SEMANTICA DE RPC EN PRESENCIA DE FALLOS

- v EL OBJETIVO DE RPC ES OCULTAR LA COMUNICACION AL HACER QUE LAS LLAMADAS A PROCEDIMIENTOS REMOTOS SE PAREZCAN A LAS LLAMADAS LOCALES.
- v EL PROBLEMA SE PRESENTA CUANDO APARECEN LOS ERRORES:
 - u LAS DIFERENCIAS ENTRE LAS LLAMADAS LOCALES Y REMOTAS NO SON TAN FACILES DE ENCUBRIR.
- v SE CONSIDERARAN LAS SIGUIENTES SITUACIONES:
 - u EL CLIENTE NO PUEDE LOCALIZAR AL SERVIDOR.
 - u SE PIERDE EL MENSAJE DE SOLICITUD DEL CLIENTE AL SERVIDOR.
 - u SE PIERDE EL MENSAJE DE RESPUESTA DEL SERVIDOR AL CLIENTE.
 - u EL SERVIDOR FALLA ANTES DE RECIBIR UNA SOLICITUD.
 - u EL CLIENTE FALLA DESPUES DE ENVIAR UNA SOLICITUD.
- v **EL CLIENTE NO PUEDE LOCALIZAR AL SERVIDOR.**
- v EL SERVIDOR PODRIA ESTAR INACTIVO.
- v EL SERVIDOR PODRIA ESTAR UTILIZANDO UNA NUEVA VERSION DE LA INTERFAZ Y NUEVOS RESGUARDOS:
 - u NO SERIAN COMPATIBLES CON LA INTERFAZ Y LOS RESGUARDOS DEL CLIENTE.
- v EN EL SERVIDOR, C / U DE LOS PROCEDIMIENTOS REGRESA UN VALOR:
 - u GENERALMENTE EL CODIGO -1 INDICA UN FALLO.
 - u TAMBIEN SE SUELE UTILIZAR UNA VARIABLE GLOBAL (UNIX) *errno* A LA QUE SE ASIGNA UN VALOR QUE INDICA EL TIPO DE ERROR.
 - u UN TIPO DE ERROR SERIA “NO SE PUDO LOCALIZAR AL SERVIDOR”.
- v OTRA POSIBILIDAD PARA EL TRATAMIENTO DE LOS ERRORES ES MEDIANTE UNA **EXCEPCION** PROVOCADA POR EL ERROR:
 - u SE CODIFICAN PROCEDIMIENTOS ESPECIALES QUE SON LLAMADOS ANTE ERRORES ESPECIFICOS.
 - u EL PROBLEMA ES QUE SE PUEDE DESTRUIR LA TRANSPARENCIA DESEADA:
 - φ SE DIFICULTA MANTENER LA SIMILITUD ENTRE PROCEDIMIENTOS LOCALES Y REMOTOS.

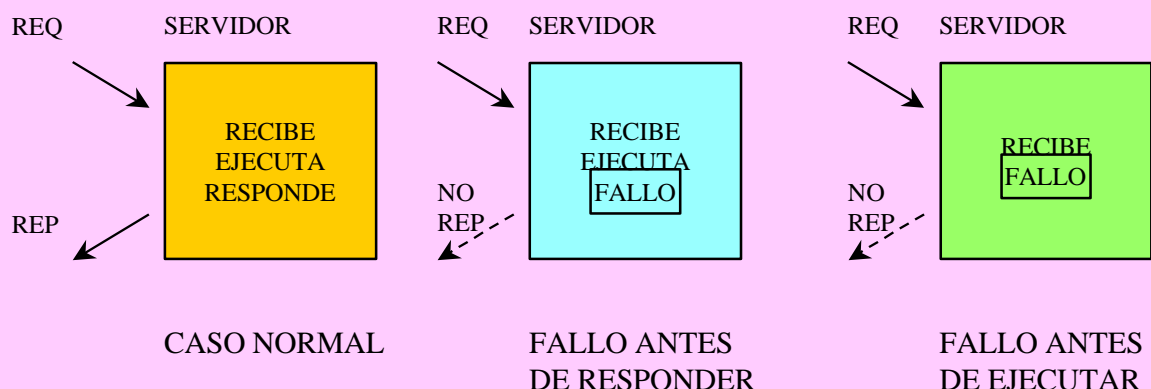
SEMANTICA DE RPC EN PRESENCIA DE FALLOS

- v **PERDIDA DE MENSAJES DE SOLICITUD.**
- v EL NUCLEO (KERNEL) DEBE INICIALIZAR UN CRONOMETRO AL ENVIAR LA SOLICITUD:
 - u SI EL TIEMPO SE TERMINA ANTES DE QUE REGRESE UNA RESPUESTA O RECONOCIMIENTO:
 - Φ EL NUCLEO VUELVE A ENVIAR EL MENSAJE.
- v SI EL MENSAJE REALMENTE SE PERDIO:
 - u EL SERVIDOR NO PODRA INDICAR LA DIFERENCIA ENTRE LA RETRANSMISION Y EL ORIGINAL Y TODO FUNCIONARA BIEN.
- v SI EL N° DE MENSAJES PERDIDOS SUPERA CIERTO LIMITE:
 - u EL NUCLEO PUEDE ASUMIR QUE EL SERVIDOR ESTA INACTIVO:
 - Φ SE REGRESA A LA SITUACION “NO SE PUDO LOCALIZAR AL SERVIDOR”.
- v **PERDIDA DE MENSAJES DE RESPUESTA.**
- v LA PERDIDA DE RESPUESTAS GENERA MAYORES PROBLEMAS QUE LA PERDIDA DE SOLICITUDES.
- v SE UTILIZA UN CRONOMETRO:
 - u SI NO LLEGA UNA RESPUESTA EN UN PERIODO RAZONABLE:
 - Φ SE DEBE VOLVER A ENVIAR LA SOLICITUD.
 - u EL PROBLEMA ES QUE EL NUCLEO DEL CLIENTE NO ESTA SEGURO DE LA RAZON POR LA QUE NO HUBO RESPUESTA.
- v CIERTAS OPERACIONES SE PUEDEN REPETIR CON SEGURIDAD TANTAS VECES COMO SEA NECESARIO SIN QUE OCURRAN DAÑOS:
 - u UNA SOLICITUD CON ESTA PROPIEDAD ES **IDEMPOTENTE**.
- v OTRAS OPERACIONES NO SON IDEMPOTENTES, POR EJ. LA TRANSFERENCIA DE DINERO:
 - u SE EMITE UNA SOLICITUD A UN SERVIDOR BANCARIO PARA TRANSFERIR CIERTA SUMA DE DINERO.
 - u LA SOLICITUD LLEGA Y SE EFECTUA PERO SE PIERDE LA RESPUESTA.
 - u EL CLIENTE CONSIDERA QUE LA SOLICITUD SE PERDIO Y LA EMITE NUEVAMENTE.
 - u EL SERVIDOR RECIBE LA NUEVA SOLICITUD Y LA EJECUTA AL NO SABER QUE ES UN REENVIO DE LA ANTERIOR.

SEMANTICA DE RPC EN PRESENCIA DE FALLOS

- v UNA FORMA DE RESOLVER EL PROBLEMA CONSISTE EN LO SIGUIENTE:
 - u EL NUCLEO DEL CLIENTE ASIGNA A CADA SOLICITUD UN N° SECUENCIAL.
 - u EL NUCLEO DEL SERVIDOR MANTIENE UN REGISTRO DEL N° SECUENCIAL DE RECEPCION MAS RECIENTE DE C / U DE LOS NUCLEOS DE CLIENTES QUE LO UTILICEN.
 - u EL NUCLEO DEL SERVIDOR PODRA INDICAR LA DIFERENCIA ENTRE UNA SOLICITUD ORIGINAL Y UNA RETRANSMISION:
 - Φ PUEDE RECHAZAR LA REALIZACION DE CUALQUIER SOLICITUD POR SEGUNDA VEZ.
- v UNA PROTECCION ADICIONAL ES TENER UN BIT EN EL ENCABEZADO DEL MENSAJE PARA DISTINGUIR LAS SOLICITUDES DE LAS RETRANSMISIONES.
- v **FALLOS DEL SERVIDOR.**
- v UN FALLO DEL SERVIDOR TAMBIEN SE RELACIONA CON LA IDEMPOTENCIA PERO NO SE PUEDE RESOLVER CON N° SECUENCIALES.

SITUACIONES POSIBLES.



SEMANTICA DE RPC EN PRESENCIA DE FALLOS

- v EL PROBLEMA ES QUE EL NUCLEO DEL CLIENTE NO PUEDE DECIDIR SI SE HA PRESENTADO LA SEGUNDA O LA TERCERA SITUACION.
- v LAS POSIBLES SOLUCIONES SON LAS SIGUIENTES:
 - u **SEMANTICA AL MENOS UNA:**
 - ⊕ ESPERAR HASTA QUE EL SERVIDOR VUELVA A ARRANCAR (O SE RECONECTE A UN NUEVO SERVIDOR) E INTENTE REALIZAR DE NUEVO LA OPERACION.
 - ⊕ MANTENER EL INTENTO HASTA RECIBIR UNA RESPUESTA PARA DARSELA AL CLIENTE.
 - ⊕ GARANTIZA QUE LA RPC SE HA REALIZADO AL MENOS UNA VEZ, PERO ES POSIBLE QUE SE REALICE MAS VECES.
 - u **SEMANTICA A LO MAS UNA:**
 - ⊕ NO SE REINTENTA Y SE INFORMA DEL FALLO.
 - ⊕ GARANTIZA QUE LA RPC SE REALIZA A LO MAS UNA VEZ, PERO ES POSIBLE QUE NO SE REALICE NI UNA SOLA VEZ.
 - u **SEMANTICA DE NO GARANTIZAR NADA:**
 - ⊕ CUANDO UN SERVIDOR FALLA, EL CLIENTE NO OBTIENE AYUDA O ALGUNA PROMESA.
 - ⊕ LA RPC SE PUEDE REALIZAR EN CUALQUIER LUGAR, UN N° DE VECES QUE VA DESDE 0 HASTA N.
 - ⊕ RESULTA FACIL DE IMPLANTAR.
 - u **SEMANTICA DE EXACTAMENTE UNA:**
 - ⊕ ES LA SOLUCION DESEABLE PERO GENERALMENTE NO EXISTE FORMA DE GARANTIZAR ESTO.
 - ⊕ EL PROCEDIMIENTO DE RECUPERACION DEPENDE TOTALMENTE DEL MOMENTO EN QUE OCURRE EL FALLO.
 - ⊕ EL CLIENTE NO TIENE FORMA DE DESCUBRIR ESE INSTANTE.
- v LA POSIBILIDAD DE FALLOS DEL SERVIDOR DISTINGUE DE MANERA CLARA LOS SISTEMAS CON UN UNICO PROCESADOR DE LOS SISTEMAS DISTRIBUIDOS:
 - u CON UN UNICO PROCESADOR EL FALLO DE UN SERVIDOR IMPLICA UN FALLO DEL CLIENTE Y LA RECUPERACION NO ES NI POSIBLE NI NECESARIA.
 - u CON SISTEMAS DISTRIBUIDOS ES POSIBLE Y NECESARIO REALIZAR CIERTA ACCION.

SEMANTICA DE RPC EN PRESENCIA DE FALLOS

- v **FALLOS DEL CLIENTE.**
- v LA CUESTION ES QUE OCURRE SI UN CLIENTE ENVIA UNA SOLICITUD A UN SERVIDOR Y FALLA ANTES DE QUE EL SERVIDOR RESPONDA.
- v SE GENERA UNA SOLICITUD DE TRABAJO O COMPUTO QUE AL FALLAR EL CLIENTE YA NADIE ESPERA:
 - u SE DICE QUE SE TIENE UN COMPUTO **HUERFANO**.
- v LOS PRINCIPALES PROBLEMAS GENERADOS POR COMPUTOS HUERFANOS SON LOS SIGUIENTES:
 - u DESPERDICIO DE CICLOS DE CPU.
 - u POSIBLE BLOQUEO DE ARCHIVOS.
 - u APROPIACION DE RECURSOS VALIOSOS.
 - u POSIBLE CONFUSION CUANDO:
 - Φ EL CLIENTE REARRANCA Y EFECTUA DE NUEVO LA RPC.
 - Φ LA RESPUESTA DEL HUERFANO REGRESA INMEDIATAMENTE LUEGO.
- v LAS SOLUCIONES A LOS COMPUTOS HUERFANOS SON LAS SIGUIENTES:
 - u **EXTERMINACION:**
 - Φ SE CREA UN REGISTRO QUE INDICA LO QUE VA A HACER EL RESGUARDO DEL CLIENTE ANTES DE QUE EMITA LA RPC.
 - Φ EL REGISTRO SE MANTIENE EN DISCO.
 - Φ LUEGO DEL REARRANQUE SE VERIFICA EL CONTENIDO DEL REGISTRO Y SE ELIMINA EL HUERFANO EXPLICITAMENTE.
 - Φ LA DESVENTAJA ES LA SOBRECARGA EN E / S GENERADA POR LA GRABACION PREVIA A CADA RPC.
 - Φ FALLARIA SI LOS HUERFANOS GENERAN RPC, CREANDO **HUERFANOS DE HUERFANOS:**
 - SERIA IMPOSIBLE LOCALIZARLOS.
 - ANTE CIERTOS FALLOS EN LA RED SERIA IMPOSIBLE ELIMINARLOS AUNQUE SE LOS LOCALICE.

SEMANTICA DE RPC EN PRESENCIA DE FALLOS

u REENCARNACION:

- ⊕ RESUELVE LOS PROBLEMAS ANTERIORES SIN NECESIDAD DE ESCRIBIR REGISTROS EN DISCO.
- ⊕ CONSISTE EN DIVIDIR EL TIEMPO EN *EPOCAS* NUMERADAS DE MANERA SECUENCIAL.
- ⊕ CUANDO UN CLIENTE REARRANCA ENVIA UN MENSAJE A TODAS LAS MAQUINAS DECLARANDO EL INICIO DE UNA NUEVA EPOCA.
- ⊕ AL RECIBIRSE ESTOS MENSAJES SE ELIMINAN TODOS LOS COMPUTOS REMOTOS.
- ⊕ SI SE DIVIDE LA RED MEDIANTE PARTICIONES POR FALLAS, PODRIAN SOBREVIVIR CIERTOS HUERFANOS:
 - CUANDO SE RECONECTEN Y VUELVAN A REPORTARSE SUS RESPUESTAS CONTENDRAN UN N° DE EPOCA OBSOLETO:
 - SE LOS PODRA DETECTAR Y ELIMINAR.

u REENCARNACION SUTIL:

- ⊕ CUANDO LLEGA UN MENSAJE DE CIERTA EPOCA:
 - CADA MAQUINA VERIFICA SI TIENE COMPUTOS REMOTOS:
 - EN CASO AFIRMATIVO INTENTA LOCALIZAR A SU POSEEDOR.
 - SI NO SE LOCALIZA AL POSEEDOR SE ELIMINA EL COMPUTO.

u EXPIRACION:

- ⊕ A CADA RPC SE LE ASIGNA UNA CANTIDAD ESTANDAR DE TIEMPO T PARA QUE REALICE SU TRABAJO.
- ⊕ SI EL TIEMPO ES INSUFICIENTE DEBE PEDIR EXPLICITAMENTE OTRO QUANTUM:
 - ESTO ES UN INCONVENIENTE.
- ⊕ SI LUEGO DEL FALLO EL SERVIDOR ESPERA T ANTES DE REARRANCAR, TODOS LOS HUERFANOS HABRAN DESAPARECIDO.
- ⊕ EL PROBLEMA ES ELEGIR UN T RAZONABLE:
 - PUEDEN EXISTIR RPC CON REQUISITOS DIVERSOS.

ASPECTOS DE LA IMPLANTACION EN RPC

- v EL DESEMPEÑO O PERFORMANCE ES FUNDAMENTAL EN LOS SISTEMAS DISTRIBUIDOS.
- v EL DESEMPEÑO DEPENDE DE MANERA CRITICA DE LA VELOCIDAD DE LA COMUNICACION.
- v LA VELOCIDAD DEPENDE EN GRAN MEDIDA DE LA IMPLANTACION.
- v **PROTOCOLOS RPC.**
- v SE DEBE ELEGIR ENTRE UN PROTOCOLO *ORIENTADO A LA CONEXION* O UN *PROTOCOLO SIN CONEXION*.
- v EN LOS PROTOCOLOS ORIENTADOS A LA CONEXION:
 - u SE ESTABLECE UNA CONEXION ENTRE CLIENTE Y SERVIDOR.
 - u TODO EL TRAFICO EN AMBAS DIRECCIONES UTILIZA ESA CONEXION.
 - u SE MANEJA A UN NIVEL INFERIOR MEDIANTE EL SOFTWARE QUE SOPORTA LA CONEXION.
 - u ES MUY UTIL PARA REDES DE AREA AMPLIA O EXTENDIDA (WAN).
 - u ES DESVENTAJOSO EN REDES DE AREA LOCAL (LAN):
 - POR LA PERDIDA DE PERFORMANCE QUE SIGNIFICARIA PROCESAR SOFTWARE ADICIONAL PARA APROVECHAR LA VENTAJA DE NO PERDER LOS PAQUETES:
 - ESTO DIFICILMENTE SE PRECISA EN LAS LAN.
 - u MUCHOS SISTEMAS DISTRIBUIDOS EN AREAS GEOGRAFICAS REDUCIDAS UTILIZAN PROTOCOLOS SIN CONEXION.
- v LOS PROTOCOLOS SIN CONEXION TIENEN CARACTERISTICAS OPUESTAS A LAS INDICADAS PRECEDENTEMENTE.
- v OTRA OPCION IMPORTANTE ES UTILIZAR UN *PROTOCOLO ESTANDAR DE PROPOSITO GENERAL* O ALGUNO *ESPECIFICO PARA RPC*.
- v LA UTILIZACION DEL PROTOCOLO ESTANDAR IP (O UDP, INTEGRADO A IP) POSEE LAS SIGUIENTES VENTAJAS:
 - u EL PROTOCOLO YA FUE DISEÑADO, LO QUE AHORRA TRABAJO.
 - u SE DISPONE DE MUCHAS IMPLANTACIONES, LO QUE TAMBIEN AHORRA TRABAJO.
 - u LOS PAQUETES IP SE PUEDEN ENVIAR Y RECIBIR POR CASI TODOS LOS SISTEMAS UNIX.
 - u LOS PAQUETES IP Y UDP SE PUEDEN TRANSMITIR EN MUCHAS DE LAS REDES EXISTENTES.

ASPECTOS DE LA IMPLANTACION EN RPC

- v EL PROBLEMA CON UN PROTOCOLO ESTANDAR TIPO IP ES EL DESEMPEÑO O PERFORMANCE:
 - u IP NO SE DISEÑO COMO UN PROTOCOLO DE USUARIO FINAL.
 - u IP SE DISEÑO COMO UNA BASE PARA QUE LAS CONEXIONES CONFIABLES TCP SE PUDIESEN ESTABLECER EN LAS INTERREDES.
 - u IP SOPORTA LA FRAGMENTACION DE PAQUETES PARA ADECUARLOS A REDES CON UN PEQUEÑO TAMAÑO MAXIMO DE PAQUETE.
 - u ESTA CARACTERISTICA DE FRAGMENTACION NO SE NECESITA EN UN SISTEMA DISTRIBUIDO BASADO EN UNA LAN, PERO:
 - ⊕ LOS CAMPOS DEL ENCABEZADO DEL PAQUETE IP RELACIONADOS CON LA FRAGMENTACION DEBEN SER:
 - LLENADOS POR EL EMISOR.
 - VERIFICADOS POR EL RECEPTOR.
 - ⊕ LOS PAQUETES IP TIENEN 13 CAMPOS DE ENCABEZADO:
 - SON UTILES SOLO 3:
 - DIRECCION FUENTE.
 - DIRECCION DESTINO.
 - LONGITUD DEL PAQUETE.
 - LOS 10 CAMPOS RESTANTES INCLUYEN UNO DE SUMA DE VERIFICACION, CUYO CALCULO CONSUME CPU.
 - UDP TIENE OTRA SUMA DE VERIFICACION, QUE TAMBIEN CUBRE LOS DATOS Y CONSUME CPU.
- v LA ALTERNATIVA ES UTILIZAR UN PROTOCOLO ESPECIALIZADO EN RPC, QUE DEBE SER INVENTADO, IMPLANTADO, PROBADO E INSERTADO EN LOS SISTEMAS EXISTENTES:
 - u DEBE SER DE ALTO RENDIMIENTO.
 - u DEBE SER ACEPTADO MASIVAMENTE.
- v OTRO ASPECTO IMPORTANTE RELACIONADO CON LOS PROTOCOLOS ES LA LONGITUD DEL PAQUETE Y EL MENSAJE:
 - u LA REALIZACION DE UNA RPC TIENE UN ALTO COSTO FIJO INDEPENDIENTE DE LA CANTIDAD DE DATOS ENVIADOS.
 - u EL PROTOCOLO Y LA RED DEBEN PERMITIR TRANSMISIONES LARGAS PARA MINIMIZAR EL N° DE RPC.

ASPECTOS DE LA IMPLANTACION EN RPC

- v **RECONOCIMIENTOS.**
- v CUANDO LOS RPC DE GRAN TAMAÑO DEBEN DIVIDIRSE EN MUCHOS PAQUETES PEQUEÑOS, SURGE LA CUESTION DEL RECONOCIMIENTO:
 - u LOS PAQUETES SERAN RECONOCIDOS *GRUPALMENTE* O *INDIVIDUALMENTE*.
 - u EJ.: UN CLIENTE DESEA ESCRIBIR UN BLOQUE DE DATOS DE 4K EN UN SERVIDOR DE ARCHIVOS, PERO EL SISTEMA NO PUEDE MANEJAR PAQUETES MAYORES DE 1K.
- v UNA ESTRATEGIA DE RECONOCIMIENTO ES EL **PROTOCOLO DETENERSE Y ESPERAR (STOP-AND-WAIT PROTOCOL):**
 - u EL CLIENTE ENVIA EL PAQUETE 0 CON EL PRIMER 1K.
 - u EL CLIENTE ESPERA UN RECONOCIMIENTO DEL SERVIDOR.
 - u EL CLIENTE ENVIA EL PAQUETE 1 CON EL SEGUNDO 1K.
 - u ETC.
 - u LA PERDIDA DE UN PAQUETE SIGNIFICA QUE NO LLEGARA SU RECONOCIMIENTO Y HABRA QUE RETRANSMITIRLO.
- v OTRA ESTRATEGIA ES EL **PROTOCOLO DE CHORRO (BLAST PROTOCOL):**
 - u EL CLIENTE ENVIA TODOS LOS PAQUETES TAN PRONTO COMO PUEDE.
 - u EL SERVIDOR RECONOCE *TODO* EL MENSAJE AL RECIBIR TODOS LOS PAQUETES:
 - Φ NO HAY RECONOCIMIENTO INDIVIDUAL DE PAQUETES.
 - u LA PERDIDA DE UN PAQUETE PUEDE SIGNIFICAR:
 - Φ LA RETRANSMISION DE TODO EL MENSAJE.
 - Φ LA **REPETICION SELECTIVA** DE LA TRANSMISION DEL PAQUETE DAÑADO O PERDIDO.
- v OTRA CONSIDERACION MAS IMPORTANTE QUE EL CONTROL DE ERRORES ES EL **CONTROL DEL FLUJO:**
 - u ESTA RELACIONADO CON LA CAPACIDAD FINITA DE RECEPCION DE PAQUETES ADYACENTES POR PARTE DE LOS CHIPS DE INTERFAZ DE RED.
 - u CUANDO UN PAQUETE LLEGA A UN RECEPTOR QUE NO LO PUEDE ACEPTAR SE PRESENTA UN **ERROR DE SOBREEJECUCION (OVERRUN ERROR)** Y EL PAQUETE SE PIERDE.

ASPECTOS DE LA IMPLANTACION EN RPC

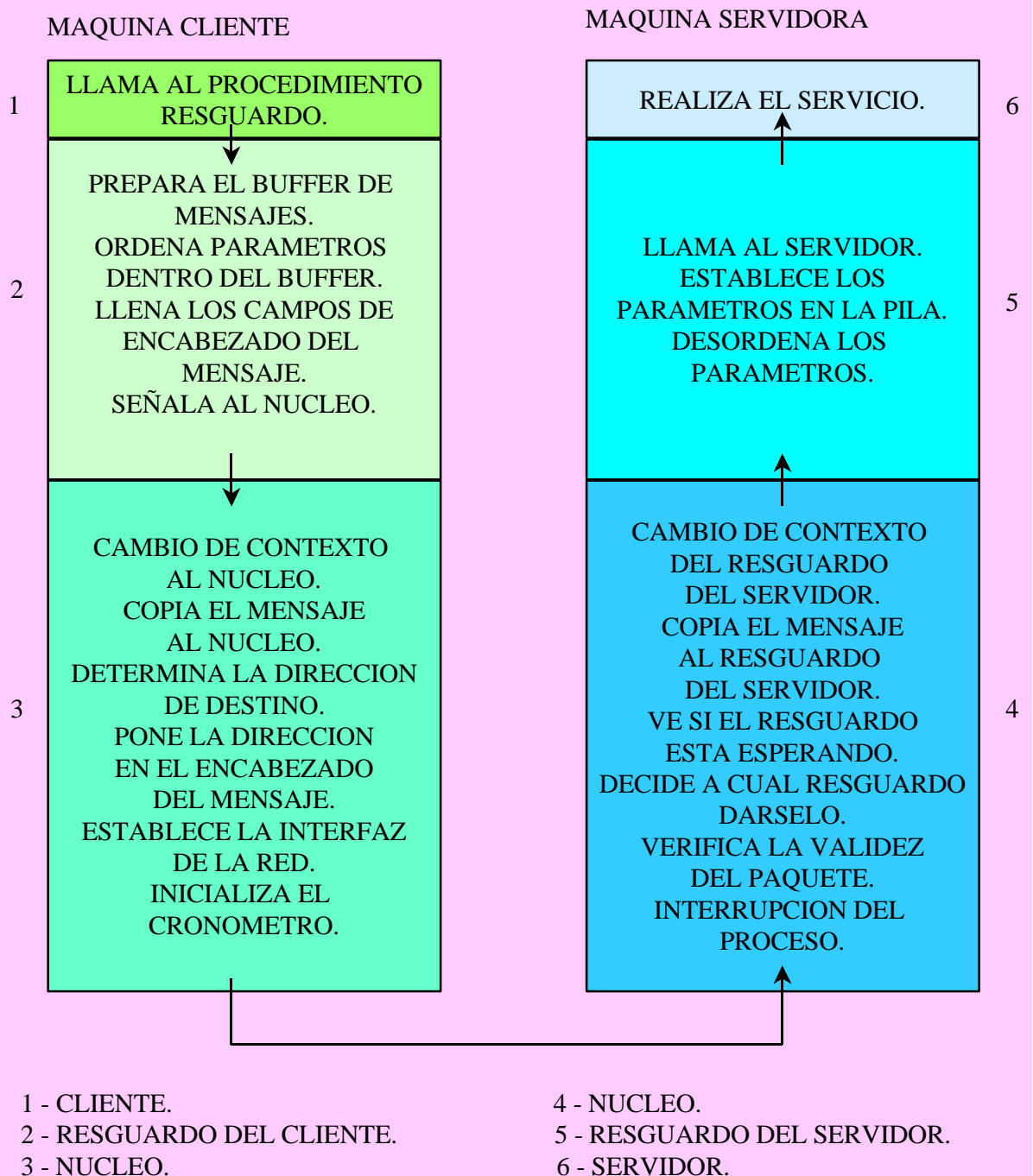
- v CON EL **PROTOCOLO DETENERSE Y ESPERAR** NO SE PRESENTAN LOS ERRORES DE SOBREEJECUCION.
- v CON EL **PROTOCOLO DE CHORRO** PUEDEN OCURRIR ERRORES DE SOBREEJECUCION.
- v UNA SOLUCION CONSISTE EN QUE EL EMISOR INSERTE UN RETRASO ENTRE LOS PAQUETES PARA DARLE TIEMPO AL RECEPTOR PARA:
 - u GENERAR LA INTERRUPCION CORRESPONDIENTE AL PAQUETE.
 - u VOLVER A ESTAR LISTO PARA RECEPCION.
- v SI LA SOBREEJECUCION SE DEBE A LA CAPACIDAD FINITA DEL BUFFER EN EL CHIP DE LA RED:
 - u EL EMISOR PUEDE:
 - Φ ENVIAR n PAQUETES Y LUEGO HACER UNA PAUSA.
 - Φ SOLICITAR UNA CONFIRMACION O RECONOCIMIENTO LUEGO DE CADA n PAQUETES.
- v UN PROBLEMA ADICIONAL CONSISTE EN LA POSIBLE PERDIDA DE PAQUETES DE RECONOCIMIENTO DEL CLIENTE AL SERVIDOR:
 - u PARA EL CLIENTE, QUE RECIBIO LA RESPUESTA A SU REQUERIMIENTO, TODO HABRA TERMINADO CORRECTAMENTE.
 - u PARA EL SERVIDOR HABRA UNA RESPUESTA NO RECONOCIDA.
 - u UNA SOLUCION ES RECONOCER TAMBIEN A LOS PAQUETES DE RECONOCIMIENTO:
 - Φ AGREGA COMPLEJIDAD Y COSTO ADICIONAL.
 - u OTRA SOLUCION ES QUE EL SERVIDOR INICIALICE UN CRONOMETRO AL ENVIAR LA RESPUESTA:
 - Φ LA RESPUESTA SE DESCARTARA CUANDO OCURRA ALGUNA DE LAS SIGUIENTES SITUACIONES:
 - LLEGADA DEL RECONOCIMIENTO.
 - EXPIRACION DEL TIEMPO.
 - LLEGADA DE UN NUEVO REQUERIMIENTO DEL CLIENTE.

ASPECTOS DE LA IMPLANTACION EN RPC

- v **RUTA CRITICA.**
- v ES LA SERIE DE INSTRUCCIONES QUE SE EJECUTAN CON CADA RPC.
- v ES IMPORTANTE DETERMINAR EN QUE PARTE DE LA RUTA CRITICA SE OCUPA LA MAYOR PARTE DEL TIEMPO QUE DEMANDA LA RPC:
 - u DEPENDE DEL TIPO DE RPC Y DE LA CANTIDAD DE DATOS QUE SE DEBEN TRANSPORTAR.
 - u EN RPC CON TRANSPORTE MINIMO LA MAYOR PARTE DEL TIEMPO SE OCUPA EN:
 - Φ EL CAMBIO DE CONTEXTO AL RESGUARDO DEL SERVIDOR AL LLEGAR UN PAQUETE.
 - Φ LA RUTINA DE SERVICIO DE INTERRUPCIONES.
 - Φ EL MOVIMIENTO DEL PAQUETE A LA INTERFAZ DE LA RED PARA SU TRANSMISION.
 - u EN RPC CON TRANSPORTE DE 1K O MAS LA MAYOR PARTE DEL TIEMPO SE OCUPA EN:
 - Φ EL TIEMPO DE TRANSMISION.
 - Φ EL TIEMPO QUE TARDA EL DESPLAZAMIENTO DEL PAQUETE HACIA ADENTRO Y AFUERA DE LA INTERFAZ.

ASPECTOS DE LA IMPLANTACION EN RPC

RUTA CRITICA DEL CLIENTE AL SERVIDOR



ASPECTOS DE LA IMPLANTACION EN RPC

- v **COPIADO.**
- v UN ASPECTO QUE DOMINA FRECUENTEMENTE LOS TIEMPOS DE EJECUCION EN RPC ES EL COPIADO.
- v EL N° DE VECES QUE SE DEBE COPIAR UN MENSAJE VARIA SEGUN:
 - u EL HARDWARE, EL SOFTWARE Y EL TIPO DE LLAMADA.
- v EN EL MEJOR DE LOS CASOS EL CHIP DE LA RED PUEDE UTILIZAR EL DMA (ACCESO DIRECTO A LA MEMORIA) PARA:
 - u TRANSFERIR EL MENSAJE DEL ESPACIO DE DIRECCIONES DEL RESGUARDO DEL CLIENTE A LA RED.
 - u DEPOSITARLO EN LA MEMORIA DEL NUCLEO DEL SERVIDOR EN TIEMPO REAL.
 - u EL NUCLEO:
 - Φ INSPECCIONA EL PAQUETE.
 - Φ ASOCIA LA PAGINA QUE LO CONTIENE EN EL ESPACIO DE DIRECCIONES DEL SERVIDOR.
 - Φ SI NO PUEDE EFECTUAR LA ASOCIACION:
 - COPIA EL PAQUETE AL RESGUARDO DEL SERVIDOR.
- v GENERALMENTE LAS COPIAS EFECTUADAS SON LAS SIGUIENTES:
 - u EL NUCLEO DEL CLIENTE COPIA EL MENSAJE DEL RESGUARDO DEL CLIENTE EN UN BUFFER DEL NUCLEO PARA SU TRANSMISION.
 - u EL NUCLEO COPIA EL MENSAJE, EN SOFTWARE, A UN BUFFER DE HARDWARE EN LA TARJETA DE INTERFAZ DE LA RED.
 - u EL PAQUETE SE DESPLAZA POR LA RED HACIA LA TARJETA DE INTERFAZ DE LA MAQUINA DESTINO.
 - u CUANDO LA INTERRUPCION CORRESPONDIENTE AL PAQUETE APARECE EN LA MAQUINA DEL SERVIDOR, EL NUCLEO LO COPIA A UN BUFFER DEL NUCLEO.
 - u EL NUCLEO DEL SERVIDOR LO EXTRAE DEL BUFFER DE HARDWARE.
 - u EL MENSAJE, LUEGO DE SER INSPECCIONADO, SE COPIA AL RESGUARDO DEL SERVIDOR.

ASPECTOS DE LA IMPLANTACION EN RPC

- v SI LA LLAMADA TRANSFIERE COMO PARAMETRO UN ARREGLO DE GRAN TAMAÑO SE AGREGAN LAS SIGUIENTES COPIAS:
 - Φ A LA PILA DEL CLIENTE PARA LA LLAMADA DEL RESGUARDO.
 - Φ DE LA PILA AL BUFFER DE MENSAJES DURANTE EL ORDENAMIENTO DENTRO DEL RESGUARDO DEL CLIENTE.
 - Φ DEL MENSAJE RECIBIDO EN EL RESGUARDO DEL SERVIDOR A LA PILA DEL SERVIDOR QUE ANTECEDE A LA LLAMADA AL SERVIDOR.
- v EL ESQUEMA PRECEDENTE INCLUYE 8 COPIAS:
 - v SI EL TIEMPO DE COPIA DE UNA PALABRA DE 32 BITS ES DE 500 NSEG, CON 8 COPIAS, CADA PALABRA NECESITA 4 MICROSEG:
 - Φ LIMITA LA VELOCIDAD MAXIMA DE TRANSMISION DE LOS DATOS A 1 MBYTE / SEG, INDEPENDIENTEMENTE DE LA VELOCIDAD DE LA RED.
- v UNA CARACTERISTICA DEL HARDWARE QUE DISMINUYE EL COPIADO ES LA **DISPERSION - ASOCIACION (SCATTER - GATHER)**:
 - v EL CHIP DE LA RED ORGANIZA UN PAQUETE CONCATENANDO 2 O MAS BUFFERS DE MEMORIA SITUADOS EN DISTINTO AMBITO DEL CLIENTE:
 - Φ NUCLEO, RESGUARDO DEL CLIENTE, ETC.
 - v EN EL SERVIDOR OCURRE ALGO SIMILAR, PERO CON MAS LIMITACIONES.
- v EN LOS S. O. CON MEMORIA VIRTUAL SE PUEDE EVITAR EL COPIADO AL RESGUARDO MEDIANTE EL SIGUIENTE PROCEDIMIENTO:
 - v EL NUCLEO MODIFICA EL MAPA DE LA MEMORIA PARA ASOCIAR EL BUFFER CON EL PAQUETE EN EL ESPACIO DE DIRECCIONES DEL SERVIDOR Y ENVIAR SIMULTANEAMENTE EL BUFFER DEL RESGUARDO DEL SERVIDOR AL NUCLEO.
 - v LOS REQUISITOS SON LOS SIGUIENTES:
 - Φ EL BUFFER DEL PAQUETE EN EL NUCLEO OCUPA TODA UNA PAGINA A PARTIR DE UNA FRONTERA DE PAGINA.
 - Φ EL BUFFER RECEPTOR DEL RESGUARDO DEL SERVIDOR TAMBIEN ES DE TODA UNA PAGINA E INICIA EN UNA FRONTERA DE PAGINA.

ASPECTOS DE LA IMPLANTACION EN RPC

- v **MANEJO DEL CRONOMETRO.**
- v LA MAYORIA DE LOS PROTOCOLOS INICIALIZAN UN CRONOMETRO CADA VEZ QUE SE ENVIA UN MENSAJE Y SE ESPERA UNA RESPUESTA.
- v SI LA RESPUESTA NO LLEGA EN EL TIEMPO ESPERADO SE VUELVE A TRANSMITIR EL MENSAJE.
- v ESTE PROCESO SE PUEDE REPETIR HASTA UN MAXIMO PREVISTO.
- v EL TIEMPO DE CPU DESTINADO AL MANEJO DE LOS CRONOMETROS PUEDE SER CONSIDERABLE.
- v EL ESTABLECIMIENTO DE UN CRONOMETRO REQUIERE CONSTRUIR UNA ESTRUCTURA DE DATOS QUE:
 - u ESPECIFIQUE EL MOMENTO EN QUE EL CRONOMETRO DEBE DETENERSE.
 - u LA ACCION A REALIZAR CUANDO ESO SUCEDA.
- v LA ESTRUCTURA DE DATOS DE UN CRONOMETRO:
 - u SE INSERTA EN UNA LISTA DE CRONOMETROS PENDIENTES CUANDO SE INICIALIZA EL CRONOMETRO.
 - u SE RETIRA DE LA LISTA CUANDO LLEGA UN RECONOCIMIENTO ANTES DE QUE TERMINE EL TIEMPO ACORDADO.
- v UN VALOR DE LAPSO DE EXPIRACION MUY PEQUEÑO HARA QUE:
 - u LOS CRONOMETROS EXPIREN CON MUCHA FRECUENCIA.
 - u SE REALICEN MUCHAS RETRANSMISIONES INNECESARIAS.
- v UN VALOR DE LAPSO DE EXPIRACION MUY GRANDE HARA QUE:
 - u SE DEMORE EN RETRANSMITIR UN PAQUETE REALMENTE PERDIDO.
- v UNA ALTERNATIVA AL ALMACENAMIENTO DE LOS CRONOMETROS EN UNA *TABLA O LISTA LIGADA ORDENADA* CONSISTE EN:
 - u UTILIZAR LA *TABLA DE PROCESOS* Y CARGAR EN ELLA UN CAMPO PARA SU TIEMPO DE EXPIRACION, SI ES QUE EXISTE.
 - u RASTREAR PERIODICAMENTE LA TABLA DE PROCESOS PARA COMPARAR EL VALOR DE CADA CRONOMETRO CON EL TIEMPO ACTUAL.
 - u ESTE TIPO DE ALGORITMOS SE DENOMINAN **ALGORITMOS DE BARRIDO (SWEEP ALGORITHMS).**

AREAS DE PROBLEMAS EN RPC

- v LA RPC MEDIANTE EL MODELO C - S SE UTILIZA AMPLIAMENTE COMO BASE DE LOS S. O. DISTRIBUIDOS.
- v LO IDEAL ES QUE LA RPC SEA **TRANSPARENTE**:
 - u EL PROGRAMADOR NO DEBE PODER DECIR SI LOS PROCEDIMIENTOS DE BIBLIOTECA SON LOCALES O REMOTOS.
 - u EL PROGRAMADOR DEBE PODER ESCRIBIR PROCEDIMIENTOS SIN IMPORTAR SI SERAN EJECUTADOS EN FORMA LOCAL O REMOTA.
 - u LA INTRODUCCION DE RPC EN UN SISTEMA QUE SE EJECUTABA ANTES EN UNA UNICA CPU NO DEBE IR ACOMPAÑADA DE UNA SERIE DE REGLAS QUE:
 - Φ PROHIBAN CONSTRUCCIONES ANTES VALIDAS.
 - Φ EXIJAN CONSTRUCCIONES QUE ANTES ERAN OPCIONALES.
- v LA MAYORIA DE LOS S. O. DISTRIBUIDOS NO CUMPLEN TOTALMENTE CON ESTOS CRITERIOS DE TRANSPARENCIA.
- v UNO DE LOS PROBLEMAS ES EL DE LAS **VARIABLES GLOBALES**:
 - u EJ.: EN UNIX EXISTE UNA VARIABLE GLOBAL *errno*:
 - Φ LUEGO DE UNA LLAMADA INCORRECTA AL SISTEMA, CONTIENE UN CODIGO QUE INDICA LO QUE ESTUVO MAL.
 - Φ SU EXISTENCIA ES INFORMACION PUBLICA, YA QUE EL ESTANDAR OFICIAL DE UNIX, POSIX, EXIGE QUE SEA VISIBLE EN UNO DE LOS ARCHIVOS DE ENCABEZADO IMPERATIVOS, *errno.h*.
 - Φ NO SE PERMITE QUE UNA IMPLANTACION LO OCULTE DE LOS PROGRAMADORES.
 - Φ SUPONGAMOS QUE UN PROGRAMADOR ESCRIBE DOS PROCEDIMIENTOS QUE TIENEN ACCESO DIRECTO A *errno*:
 - UNO SE EJECUTA EN FORMA LOCAL, EL OTRO EN FORMA REMOTA.
 - UNO DE LOS PROCEDIMIENTOS TENDRA UN ACCESO INCORRECTO.
 - Φ EL PROBLEMA ES QUE NO SE PUEDE IMPLANTAR EL PERMISO PARA EL ACCESO IRRESTRICTO DE LOS PROCEDIMIENTOS LOCALES A LAS VARIABLES GLOBALES REMOTAS Y VICEVERSA.

AREAS DE PROBLEMAS EN RPC

- LA PROHIBICION DEL ACCESO IRRESTRICTO MENCIONADO VIOLA EL PRINCIPIO DE TRANSPARENCIA:
 - LOS PROGRAMAS NO DEBEN ACTUAR DE MANERA DISTINTA SEGUN RPC.
- v OTRO PROBLEMA SON LOS **LENGUAJES DEBILMENTE TIPIFICADOS**, COMO POR EJEMPLO, “C”:
 - EN UN LENGUAJE FUERTEMENTE TIPIFICADO, COMO “PASCAL”:
 - EL COMPILADOR Y EL PROCEDIMIENTO RESGUARDO CONOCEN TODO LO RELATIVO A TODOS LOS PARAMETROS.
 - EL RESGUARDO PUEDE ORDENAR LOS PARAMETROS SIN DIFICULTAD.
 - CON “C” SE PUEDE TRABAJAR CON ARREGLOS SIN ESPECIFICAR SU TAMAÑO:
 - EL RESGUARDO DEL CLIENTE NO PUEDE ORDENAR LOS PARAMETROS:
 - NO PUEDE DETERMINAR SU TAMAÑO.
 - UNA SOLUCION CONSISTE EN QUE EL PROGRAMADOR DEFINA EL TAMAÑO MAXIMO CUANDO ESCRIBA LA DEFINICION FORMAL DEL SERVIDOR:
 - RESTA FLEXIBILIDAD Y PUEDE GENERAR PROBLEMAS DE MEMORIA.
- v UN PROBLEMA ADICIONAL CONSISTE EN QUE **NO SIEMPRE ES POSIBLE DEDUCIR LOS TIPOS DE LOS PARAMETROS**:
 - NI SIQUIERA A PARTIR DE UNA ESPECIFICACION FORMAL DEL PROPIO CODIGO, EN ESPECIAL CONSIDERANDO “C”.
 - LA EXCLUSION DE “C” CUANDO SE UTILICE RPC VIOLARIA LA TRANSPARENCIA.
- v **CONCLUSION**:
 - *EL MODELO C - S SE AJUSTA A MUCHOS CASOS PERO NO ES PERFECTO.*

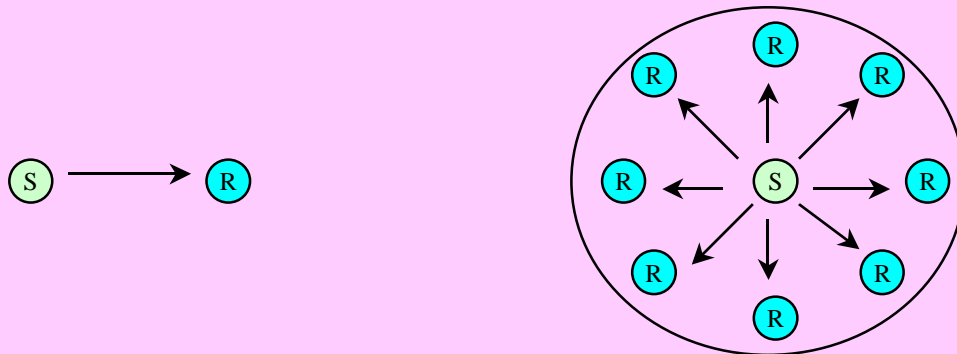
COMUNICACION EN GRUPO

- v UNA HIPOTESIS SUBYACENTE E INTRINSECA DE RPC ES QUE LA COMUNICACION SOLO ES ENTRE *DOS* PARTES:
 - u EL *CLIENTE* Y EL *SERVIDOR*.
- v A VECES EXISTEN CIRCUNSTANCIAS EN LAS QUE LA COMUNICACION ES ENTRE *VARIOS* PROCESOS Y NO SOLO DOS:
 - u EJ.: UN GRUPO DE SERVIDORES DE ARCHIVO QUE COOPERAN PARA OFRECER UN UNICO SERVICIO DE ARCHIVOS TOLERANTE A FALLOS:
 - Φ SERIA RECOMENDABLE QUE UN CLIENTE ENVIE EL MENSAJE A TODOS LOS SERVIDORES PARA GARANTIZAR LA EJECUCION DE LA SOLICITUD AUNQUE ALGUNO FALLE.
 - u RPC NO PUEDE CONTROLAR LA COMUNICACION DE UN SERVIDOR CON MUCHOS RECEPTORES, A MENOS QUE REALICE RPC CON C / U EN FORMA INDIVIDUAL.
- v UN *GRUPO* ES UNA COLECCION DE PROCESOS QUE ACTUAN JUNTOS EN CIERTO SISTEMA O ALGUNA FORMA DETERMINADA POR EL USUARIO.
- v LA PROPIEDAD FUNDAMENTAL DE TODOS LOS GRUPOS ES QUE CUANDO UN MENSAJE SE ENVIA AL PROPIO GRUPO:
 - u TODOS LOS MIEMBROS DEL GRUPO LO RECIBEN.
- v SE TRATA DE UNA COMUNICACION **UNO - MUCHOS (UN EMISOR, MUCHOS RECEPTORES)**:
 - u SE DISTINGUE DE LA COMUNICACION **PUNTUAL O PUNTO A PUNTO (UN EMISOR, UN RECEPTOR)**.
- v LOS GRUPOS SON DINAMICOS:
 - u SE PUEDEN CREAR Y DESTRUIR.
 - u UN PROCESO SE PUEDE UNIR A UN GRUPO O DEJAR A OTRO.
 - u UN PROCESO PUEDE SER MIEMBRO DE VARIOS GRUPOS A LA VEZ.
- v LA IMPLANTACION DE LA COMUNICACION EN GRUPO DEPENDE EN GRAN MEDIDA DEL HARDWARE:
 - u EN CIERTAS REDES ES POSIBLE CREAR UNA DIRECCION ESPECIAL DE RED A LA QUE PUEDEN ESCUCHAR VARIAS MAQUINAS:
 - Φ CUANDO SE ENVIA UN MENSAJE A UNA DE ESAS DIRECCIONES SE LO ENTREGA AUTOMATICAMENTE A TODAS LAS MAQUINAS QUE ESCUCHAN A ESA DIRECCION.
 - Φ ESTA TECNICA SE DENOMINA **MULTITRANSMISION**.
 - Φ CADA GRUPO DEBE TENER UNA DIRECCION DE MULTITRANSMISION DISTINTA.

COMUNICACION EN GRUPO

- v LAS REDES QUE NO SOPORTAN MULTITRANSMISION OPERAN CON **TRANSMISION SIMPLE**:
 - v SIGNIFICA QUE LOS PAQUETES QUE TIENEN CIERTA DIRECCION SE ENTREGAN A TODAS LAS MAQUINAS.
 - v SE PUEDE UTILIZAR PARA IMPLANTAR LOS GRUPOS, PERO ES MENOS EFICIENTE QUE LA MULTITRANSMISION.
 - v CADA MAQUINA DEBE VERIFICAR, MEDIANTE SU SOFTWARE, SI EL PAQUETE VA DIRIGIDO A ELLA:
 - Φ EN CASO NEGATIVO SE DESCARTA, PERO PARA ANALIZARLO SE GENERO UNA INTERRUPCION Y SE DEDICO CICLOS DE CPU.
- v OTRA SOLUCION ES IMPLANTAR LA COMUNICACION EN GRUPO MEDIANTE LA TRANSMISION POR PARTE DEL EMISOR DE **PAQUETES INDIVIDUALES** A C / U DE LOS MIEMBROS DEL GRUPO:
 - v EN VEZ DE UN PAQUETE SE PRECISAN n PAQUETES.
 - v ES MENOS EFICIENTE QUE LAS SOLUCIONES ANTERIORES.
 - v ES UNA SOLUCION VALIDA PARTICULARMENTE CON GRUPOS PEQUEÑOS.
 - v EL ENVIO DE UN MENSAJE DE UN EMISOR A UN UNICO RECEPTOR SE LLAMA **UNITRANSMISION**.

COMUNICACION PUNTO A PUNTO Y COMUNICACION UNO A MUCHOS



ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v EN LA COMUNICACION EN GRUPO TAMBIEN SE PRESENTAN POSIBILIDADES TALES COMO:
 - u ALMACENAMIENTO EN BUFFERS VS. EL NO ALMACENAMIENTO.
 - u BLOQUEO VS. NO BLOQUEO.
- v ADEMAS EXISTEN OTRAS POSIBILIDADES QUE NO SE DAN EN LA COMUNICACION ENTRE UN EMISOR Y UN SOLO RECEPTOR.
- v **GRUPOS CERRADOS VS. GRUPOS ABIERTOS.**
- v EN LOS **GRUPOS CERRADOS**:
 - u SOLO LOS MIEMBROS DEL GRUPO PUEDEN ENVIAR HACIA EL GRUPO.
 - u LOS EXTRAÑOS NO PUEDEN ENVIAR MENSAJES AL GRUPO COMO UN TODO:
 - Φ PUEDEN ENVIAR MENSAJES A MIEMBROS DEL GRUPO EN LO INDIVIDUAL.
- v EN LOS **GRUPOS ABIERTOS**:
 - u CUALQUIER PROCESO DEL SISTEMA PUEDE ENVIAR A CUALQUIER GRUPO.
- v LOS GRUPOS CERRADOS SE UTILIZAN GENERALMENTE PARA EL PROCESAMIENTO PARALELO:
 - u EJ.: UN CONJUNTO DE PROCESOS QUE TRABAJAN DE MANERA CONJUNTA, TIENEN SU PROPIO OBJETIVO Y NO INTERACTUAN CON EL MUNDO EXTERIOR.
- v CUANDO LA IDEA DE GRUPO PRETENDE SOPORTAR SERVIDORES DUPLICADOS:
 - u ES IMPORTANTE QUE LOS PROCESOS QUE NO SEAN MIEMBROS (CLIENTES) PUEDAN ENVIAR HACIA EL GRUPO.
 - u PODRIA SER NECESARIO QUE LOS MIEMBROS DEL GRUPO UTILIZARAN LA COMUNICACION EN GRUPO.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v **GRUPOS DE COMPAÑEROS VS. GRUPOS JERARQUICOS.**
- v EN ALGUNOS GRUPOS TODOS LOS PROCESOS SON IGUALES:
 - u NO EXISTE DISTINCION DE JERARQUIAS.
 - u SON LOS **GRUPOS DE COMPAÑEROS.**
- v EN OTROS GRUPOS EXISTE CIERTO TIPO DE JERARQUIA:
 - u SON LOS **GRUPOS JERARQUICOS.**
 - u EJ.: UN PROCESO ES EL COORDINADOR Y TODOS LOS DEMAS SON LOS TRABAJADORES.
 - u UNA SOLICITUD DE UN TRABAJO QUE SE GENERE EN UN CLIENTE EXTERNO O EN UNO DE LOS PROCESOS TRABAJADORES:
 - Φ SE ENVIA AL COORDINADOR.
 - Φ EL COORDINADOR DECIDE CUAL DE LOS TRABAJADORES ES EL MAS ADECUADO PARA LLEVARLA A CABO Y SE LA ENVIA.
- v CADA TIPO DE GRUPO TIENE SUS VENTAJAS Y DESVENTAJAS:
 - u RESPECTO DEL GRUPO DE COMPAÑEROS:
 - Φ ES SIMETRICO Y NO TIENE UN PUNTO DE FALLO.
 - Φ SI UNO DE LOS PROCESOS FALLA, EL GRUPO SE REDUCE PERO PUEDE CONTINUAR.
 - Φ PARA TOMAR DECISIONES GRUPALES SE PRODUCEN RETRASOS DEBIDOS A LA COMUNICACION ENTRE LOS MIEMBROS DEL GRUPO.
 - u RESPECTO DEL GRUPO JERARQUICO:
 - Φ LA PERDIDA DEL COORDINADOR LLEVA AL GRUPO A UN ALTO TOTAL, LO QUE ES UNA DESVENTAJA.
 - Φ EN CONDICIONES NORMALES, EL COORDINADOR FUNCIONA CORRECTAMENTE Y TOMA DECISIONES SIN MOLESTAR A LOS DEMAS PROCESOS.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v UN EJ. DE GRUPO JERARQUICO PODRIA SER UN PROGRAMA DE AJEDREZ EN PARALELO:
 - u EL COORDINADO:
 - Φ TOMA EL TABLERO ACTUAL.
 - Φ GENERA TODOS LOS MOVIMIENTOS VALIDOS A PARTIR DE EL.
 - Φ LOS ENVIA A LOS TRABAJADORES PARA SU EVALUACION.
 - Φ CONTROLA LA ESTRATEGIA DE BUSQUEDA.
 - Φ DESARROLLA EL ARBOL DEL JUEGO.
 - u LOS TRABAJADORES:
 - Φ AL PROCESAR LAS EVALUACIONES GENERAN NUEVOS TABLEROS.
 - Φ LOS TABLEROS SE ENVIAN AL COORDINADOR.
 - Φ AL QUEDAR INACTIVOS, SOLICITAN AL COORDINADOR UN NUEVO TABLERO EN EL CUAL TRABAJAR.
- v **MEMBRESIA DEL GRUPO.**
- v LA COMUNICACION EN GRUPO REQUIERE CIERTO METODO PARA:
 - u CREACION Y ELIMINACION DE GRUPOS.
 - u UNION Y SEPARACION DE PROCESOS A GRUPOS.
- v UNA POSIBILIDAD ES TENER UN **SERVIDOR DE GRUPOS** AL CUAL ENVIAR TODAS LAS SOLICITUDES:
 - u ES UN METODO DIRECTO, EFICIENTE Y FACIL DE IMPLEMENTAR.
 - u LA DESVENTAJA ES EL PUNTO DE FALLO QUE REPRESENTA LA ADMINISTRACION CENTRALIZADA DE LOS GRUPOS.
- v OTRA POSIBILIDAD ES LA **ADMINISTRACION DISTRIBUIDA** DE LAS MEMBRESIAS A GRUPOS:
 - u EN UN GRUPO ABIERTO, UN PROCESO EXTRAÑO PUEDE ENVIAR UN MENSAJE A LOS INTEGRANTES DEL GRUPO PARA ANUNCIAR SU PRESENCIA.
 - u EN UN GRUPO CERRADO SE PRECISA ALGO SIMILAR, YA QUE SE DEBE CONTEMPLAR LA ADMISION DE NUEVOS MIEMBROS AL GRUPO CERRADO.
 - u AL SALIR DE UN GRUPO, EL PROCESO DEBE COMUNICARLO A LOS DEMAS DEL GRUPO QUE DEJA.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v UN ASPECTO PROBLEMÁTICO SE PRESENTA CUANDO UN MIEMBRO FALLA, SALIENDO POR LO TANTO DEL GRUPO:
 - u NO HAY UN ANUNCIO APROPIADO DE ESTE HECHO.
 - u LOS DEMÁS MIEMBROS DEL GRUPO LO DEBEN DESCUBRIR DE FORMA EXPERIMENTAL:
 - Φ LUEGO SE LO PUEDE ELIMINAR DEL GRUPO.
- v OTRO ASPECTO IMPORTANTE ES QUE LA ENTRADA Y SALIDA AL GRUPO DEBE SINCRONIZARSE CON EL ENVÍO DE MENSAJES:
 - u UN PROCESO QUE SE UNIO A UN GRUPO DEBE RECIBIR TODOS LOS MENSAJES QUE SE ENVÍEN AL MISMO.
 - u UN PROCESO QUE HA SALIDO DE UN GRUPO:
 - Φ NO DEBE RECIBIR MÁS MENSAJES DEL GRUPO.
 - Φ EL GRUPO NO DEBE RECIBIR MÁS MENSAJES DEL PROCESO.
 - Φ LOS OTROS MIEMBROS NO DEBEN RECIBIR MÁS MENSAJES DEL PROCESO SALIENTE.
 - u UNA FORMA DE GARANTIZAR QUE UNA ENTRADA O SALIDA SE INTEGRA AL FLUJO DE MENSAJES EN EL LUGAR CORRECTO ES:
 - Φ CONVERTIR ESTA OPERACIÓN EN UN MENSAJE A TODO EL GRUPO.
- v UN ASPECTO CRÍTICO RESULTA CUANDO FALLAN TANTAS MÁQUINAS QUE EL GRUPO YA NO PUEDE FUNCIONAR:
 - u SE NECESITA CIERTO PROTOCOLO PARA RECONSTRUIR EL GRUPO.
 - u ALGUNO DE LOS PROCESOS DEBERÁ TOMAR LA INICIATIVA.
 - u EL PROTOCOLO DEBERÁ RESOLVER LA SITUACIÓN QUE SE PRESENTA CUANDO DOS O MÁS PROCESOS INTENTAN AL MISMO TIEMPO RECONSTRUIR EL GRUPO.
- v **DIRECCIONAMIENTO AL GRUPO.**
- v LOS GRUPOS DEBEN PODER DIRECCIONARSE, AL IGUAL QUE LOS PROCESOS.
- v UNA FORMA ES DARLE A CADA GRUPO UNA DIRECCIÓN ÚNICA, SIMILAR A UNA DIRECCIÓN DE PROCESO.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v SI LA RED SOPORTA MULTITRANSMISION:
 - u LA DIRECCION DEL GRUPO SE PUEDE ASOCIAR CON UNA DIRECCION DE MULTITRANSMISION.
 - u CADA MENSAJE ENVIADO A LA DIRECCION DEL GRUPO SE PODRA MULTITRANSMITIR.
- v SI LA RED NO SOPORTA MULTITRANSMISION:
 - u SE TENDRA QUE UTILIZAR TRANSMISION SIMPLE.
 - u CADA NUCLEO LO RECIBIRA Y EXTRAERA LA DIRECCION DEL GRUPO.
 - u SI NINGUNO DE LOS PROCESOS EN LA MAQUINA ES UN MIEMBRO DEL GRUPO, SE DESCARTA LA TRANSMISION.
 - u EN CASO CONTRARIO SE TRANSFIERE A TODOS LOS MIEMBROS DEL GRUPO.
- v SI LA RED NO SOPORTA MULTITRANSMISION NI TRANSMISION SIMPLE:
 - u SE TENDRA QUE UTILIZAR UNITRANSMISION.
 - u EL NUCLEO DE LA MAQUINA EMISORA DEBERA CONTAR CON UNA LISTA DE LAS MAQUINAS QUE TIENEN PROCESOS PERTENECIENTES AL GRUPO.
 - u DEBERA ENVIAR A CADA MAQUINA UN MENSAJE PUNTUAL.
- v UN SEGUNDO METODO DE DIRECCIONAMIENTO DE GRUPO CONSISTE EN PEDIRLE AL EMISOR UNA LISTA EXPLICITA DE TODOS LOS DESTINOS:
 - u EJ.: LISTA DE DIRECCIONES IP.
 - u EL PARAMETRO DE LA LLAMADA *send* QUE ESPECIFICA EL DESTINO ES UN APUNTADOR A UNA LISTA DE DIRECCIONES.
 - u LA DESVENTAJA CONSISTE EN QUE LOS PROCESOS DEL USUARIO (LOS MIEMBROS DEL GRUPO) DEBEN TENER CONOCIMIENTO DE QUIEN ES MIEMBRO DE CADA GRUPO:
 - Φ NO ES TRANSPARENTE.
 - Φ LOS PROCESOS DEL USUARIO DEBEN ACTUALIZAR LAS LISTAS DE MIEMBROS.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v UN TERCER METODO ES EL DE **DIRECCIONAMIENTO DE PREDICADOS (PREDICATE ADDRESSING)**:
 - u EL MENSAJE SE ENVIA A TODOS LOS MIEMBROS DEL GRUPO (O A TODO EL SISTEMA) MEDIANTE UNO DE LOS METODOS ANTERIORES.
 - u EL MENSAJE CONTIENE UN PREDICADO (EXPRESION BOOLEANA) PARA SER EVALUADO.
 - u EL PREDICADO PUEDE UTILIZAR EL N° DE MAQUINA DEL RECEPTOR, SUS VARIABLES LOCALES U OTROS FACTORES.
 - u SI EL VALOR DEL PREDICADO ES “VERDADERO” SE ACEPTA EL MENSAJE Y SE DESCARTA SI ES “FALSO”.
 - u PERMITE ENVIAR UN MENSAJE SOLO A AQUELLAS MAQUINAS QUE TENGAN AL MENOS X MB DE MEMORIA LIBRE Y SE PUEDAN OCUPAR DE UN NUEVO PROCESO.
- v **PRIMITIVAS SEND Y RECEIVE.**
- v EL ENVIO DE UN MENSAJE A UN GRUPO NO SE PUEDE MODELAR COMO UNA LLAMADA A UN PROCEDIMIENTO.
- v CON LA COMUNICACION EN GRUPO EXISTEN EN POTENCIA n RESPUESTAS DIFERENTES:
 - u NO RESULTA APLICABLE EL ESQUEMA DE RPC.
- v SE UTILIZAN LLAMADAS EXPLICITAS PARA EL ENVIO Y RECEPCION (MODELO DE UN SOLO SENTIDO).
- v SI SE UNIFICAN LAS PRIMITIVAS PUNTUALES Y GRUPALES PARA *send*:
 - u UNO DE LOS PARAMETROS INDICA EL DESTINO:
 - Φ SI ES UNA DIRECCION DE UN PROCESO, SE ENVIA UN UNICO MENSAJE A ESE PROCESO EN PARTICULAR.
 - Φ SI ES UNA DIRECCION DE GRUPO (O UN APUNTADOR A UNA LISTA DE DESTINOS), SE ENVIA UN MENSAJE A TODOS LOS MIEMBROS DEL GRUPO.
 - u UN SEGUNDO PARAMETRO APUNTA AL MENSAJE POR ENVIAR.
- v SI SE FUSIONAN LAS PRIMITIVAS PUNTUALES Y GRUPALES PARA *receive*:
 - u LA OPERACION CONCLUYE CUANDO LLEGA UN MENSAJE PUNTUAL O UN MENSAJE DE GRUPO.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v SI ES NECESARIO QUE LAS RESPUESTAS ESTEN ASOCIADAS A SOLICITUDES PREVIAS:
 - υ SE ENVIA UN MENSAJE.
 - υ SE EFECTUA VARIAS VECES UN PROCESO *get_reply* PARA RECOLECTAR TODAS LAS RESPUESTAS, UNA A LA VEZ.
- v **ATOMICIDAD.**
- v LA MAYORIA DE LOS SISTEMAS DE COMUNICACION EN GRUPO ESTAN DISEÑADOS PARA QUE LOS MENSAJES ENVIADOS AL GRUPO LLEGUEN CORRECTAMENTE A TODOS LOS MIEMBROS O A NINGUNO DE ELLOS:
 - υ ESTA PROPIEDAD DE “TODO O NADA” EN LA ENTREGA SE LLAMA **ATOMICIDAD O TRANSMISION ATOMICA.**
 - υ FACILITA LA PROGRAMACION DE LOS SISTEMAS DISTRIBUIDOS.
 - υ ES DE GRAN IMPORTANCIA PARA GARANTIZAR LA CONSISTENCIA DE LAS BASES DE DATOS Y DE LOS ARCHIVOS DISTRIBUIDOS Y DUPLICADOS.
- v LA UNICA FORMA DE GARANTIZAR QUE CADA DESTINO RECIBE TODOS SUS MENSAJES ES PEDIRLE QUE ENVIE DE REGRESO UN RECONOCIMIENTO DESPUES DE RECIBIR EL MENSAJE:
 - υ ESTO FUNCIONA SI LAS MAQUINAS NO FALLAN.
 - υ SI FALLAN:
 - ⊕ ALGUNOS MIEMBROS DEL GRUPO HABRAN RECIBIDO EL MENSAJE Y OTROS NO; ESTO ES INACEPTABLE.
 - ⊕ LOS MIEMBROS QUE NO RECIBIERON EL MENSAJE NI SIQUIERA SABEN QUE LES FALTA ALGO, POR LO QUE NO PEDIRAN UNA RETRANSMISION:
 - SI PUDIERAN DETECTAR EL FALTANTE PERO FALLARA EL EMISOR, NO PODRAN RECIBIR EL MENSAJE.
- v UNA SOLUCION PUEDE LLEGAR DEL **ALGORITMO DE JOSEPH Y BIRMAN:**
 - υ DEMUESTRA LA POSIBILIDAD DE LA TRANSMISION ATOMICA.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- u EL EMISOR COMIENZA CON EL ENVIO DE UN MENSAJE A TODOS LOS MIEMBROS DEL GRUPO.
- u LOS CRONOMETROS SE ACTIVAN Y SE ENVIAN LAS RETRANSMISIONES EN LOS CASOS NECESARIOS.
- u CUANDO UN PROCESO RECIBE UN MENSAJE:
 - Φ SI NO RECIBIO YA ESTE MENSAJE PARTICULAR:
 - LO ENVIA A TODOS LOS MIEMBROS DEL GRUPO:
 - CON CRONOMETROS Y RETRANSMISIONES EN LOS CASOS NECESARIOS.
 - Φ SI YA RECIBIO ESTE MENSAJE PARTICULAR:
 - NO SE EFECTUAN ENVIOS Y EL MENSAJE SE DESCARTA.
- v ESTE ALGORITMO ASEGURA QUE TODOS LOS PROCESOS SOBREVIVIENTES OBTENDRAN EL MENSAJE:
 - u INDEPENDIENTEMENTE DEL N° DE MAQUINAS QUE FALLEN O EL N° DE PAQUETES PERDIDOS.
- v **ORDENAMIENTO DE MENSAJES.**
- v EL ORDENAMIENTO DE LOS MENSAJES ES UN ASPECTO FUNDAMENTAL EN LA COMUNICACION EN GRUPO.
- v EJ.: CONSIDERAMOS 5 MAQUINAS, C / U CON UN PROCESO:
 - u LOS PROCESOS SE IDENTIFICAN COMO 0, 1, 2, 3 Y 4.
 - u LOS PROCESOS 0, 1, 3 Y 4 PERTENECEN AL MISMO GRUPO.
 - u LOS PROCESOS 0 Y 4 DESEAN ENVIAR UN MENSAJE AL GRUPO SIMULTANEAMENTE:
 - Φ SUPONGAMOS QUE NO SE DISPONE DE MULTITRANSMISION NI DE TRANSMISION SIMPLE.
 - Φ CADA PROCESO DEBE ENVIAR 3 MENSAJES INDEPENDIENTES (UNITRANSMISION).
 - u EL PROCESO 0 ENVIA A LOS PROCESOS 1, 3 Y 4.
 - u EL PROCESO 4 ENVIA A LOS PROCESOS 0, 1 Y 3.
 - u UNA POSIBLE SECUENCIA DE INTERCALACION DE LOS MENSAJES ES LA SIGUIENTE:
 - Φ 0 A 1; 4 A 0; 4 A 1; 4 A 3; 0 A 3; 0 A 4.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v EL PROBLEMA ES QUE CUANDO DOS PROCESOS CONTIENDEN POR EL ACCESO A UNA LAN, *EL ORDEN DE ENVIO DE LOS MENSAJES NO ES DETERMINISTA.*
- v EN EL EJEMPLO ANTERIOR, LOS PROCESOS 1 Y 3 RECIBEN LOS MENSAJES DE LOS PROCESOS 0 Y 4 EN DISTINTO ORDEN:
 - v SI LOS PROCESOS 0 Y 4 INTENTAN ACTUALIZAR EL MISMO REGISTRO DE UNA BASE DE DATOS:
 - ⊕ LOS PROCESOS 1 Y 3 TERMINARAN CON DISTINTOS VALORES FINALES.
- v *UN SISTEMA DEBE TENER UNA SEMANTICA BIEN DEFINIDA CON RESPECTO AL ORDEN DE ENTREGA DE LOS MENSAJES.*
- v LA MEJOR GARANTIA ES LA ENTREGA INMEDIATA DE TODOS LOS MENSAJES, EN EL ORDEN EN QUE FUERON ENVIADOS:
 - v TODOS LOS MENSAJES DESTINADOS A UN GRUPO DEBEN ENTREGARSE ANTES DE COMENZAR A ENTREGAR LA SIGUIENTE SERIE DE MENSAJES.
 - v TODOS LOS RECEPTORES RECIBEN TODOS LOS MENSAJES EN EL MISMO ORDEN.
 - v ES ESQUEMA SE DENOMINA **ORDENAMIENTO CON RESPECTO AL TIEMPO GLOBAL.**
- v UNA VARIANTE AL ESQUEMA ANTERIOR ES EL **ORDENAMIENTO CONSISTENTE:**
 - v SI DOS MENSAJES A Y B SE ENVIAN MUY CERCANOS EN EL TIEMPO, EL SISTEMA:
 - ⊕ ELIGE UNO DE ELLOS COMO EL “PRIMERO”.
 - ⊕ LO ENVIA A TODOS LOS MIEMBROS DEL GRUPO.
 - ⊕ LUEGO ENVIA EL OTRO MENSAJE.
 - v SE GARANTIZA QUE LOS MENSAJES LLEGUEN A TODOS LOS MIEMBROS DEL GRUPO EN EL MISMO ORDEN:
 - ⊕ DICHO ORDEN PODRIA NO SER AQUEL CON EL QUE FUERON ENVIADOS.

ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO

- v **GRUPOS TRASLAPADOS.**
- v UN PROCESO PUEDE SER MIEMBRO DE VARIOS GRUPOS A LA VEZ:
 - u ESTO PUEDE GENERAR UN NUEVO TIPO DE INCONSISTENCIA.
- v EJ.: SUPONGAMOS QUE:
 - u EL GRUPO 1 ESTA FORMADO POR LOS PROCESOS *A, B Y C*.
 - u EL GRUPO 2 ESTA FORMADO POR LOS PROCESOS *B, C Y D*.
 - u *C / U* DE LOS PROCESOS *A Y D* DECIDE DE MANERA SIMULTANEA ENVIAR UN MENSAJE A SUS GRUPOS RESPECTIVOS.
 - u EL SISTEMA UTILIZA EL ORDENAMIENTO CON RESPECTO AL TIEMPO GLOBAL DENTRO DE CADA GRUPO.
 - u SE UTILIZA LA UNITRANSMISION.
 - u EL ORDEN DE LOS MENSAJES ES EL SIGUIENTE:
 - Φ *A A B; D A B; D A C Y A A C*.
 - Φ SE DA LA SITUACION DE QUE DOS PROCESOS, *B Y C*, RECIBEN LOS MENSAJES EN UN ORDEN DISTINTO.
- v EL PROBLEMA ES EL SIGUIENTE:
 - u EXISTE UN ORDENAMIENTO CON RESPECTO AL TIEMPO GLOBAL DENTRO DE CADA GRUPO.
 - u NO EXISTE COORDINACION ENTRE VARIOS GRUPOS.
 - u RESULTA MUY COMPLICADO IMPLANTAR EL ORDEN CON RESPECTO AL TIEMPO ENTRE LOS DISTINTOS GRUPOS:
 - Φ NO SIEMPRE ES NECESARIO HACERLO.
- v **ESCALABILIDAD.**
- v ES NECESARIO CONSIDERAR COMO FUNCIONARAN LOS ALGORITMOS CUANDO SE TIENEN LOS SIGUIENTES CASOS:
 - u GRUPOS CON CENTENAS O MILES DE MIEMBROS.
 - u CENTENAS O MILES DE GRUPOS.
 - u UTILIZACION DE VARIAS LAN Y COMPUERTAS (GATEWAYS) CONECTADAS, ES DECIR INTERRED (INTERNETWORK).
 - u DISEMINACION DE GRUPOS EN VARIOS CONTINENTES.
- v LAS COMPUERTAS PUEDEN AFECTAR LA MULTITRANSMISION Y EL HECHO REQUERIDO POR ALGUNAS REDES DE TENER UN SOLO PAQUETE EN LA RED EN UN INSTANTE DADO.

SINCRONIZACION EN SISTEMAS DISTRIBUIDOS

- v INTRODUCCION A LA SINCRONIZACION EN SISTEMAS DISTRIBUIDOS
- v SINCRONIZACION DE RELOJES
- v RELOJES LOGICOS
- v RELOJES FISICOS
- v ALGORITMOS PARA LA SINCRONIZACION DE RELOJES
- v EXCLUSION MUTUA
- v ALGORITMOS DE ELECCION
- v TRANSACCIONES ATOMICAS
- v EL MODELO DE TRANSACCION
- v IMPLANTACION DEL MODELO DE TRANSACCION
- v CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION
- v BLOQUEOS EN SISTEMAS DISTRIBUIDOS
- v DETECCION DISTRIBUIDA DE BLOQUEOS
- v PREVENCIÓN DISTRIBUIDA DE BLOQUEOS

INTRODUCCION A LA SINCRONIZACION EN SISTEMAS DISTRIBUIDOS

- v ADEMAS DE LA COMUNICACION, ES FUNDAMENTAL LA FORMA EN QUE LOS PROCESOS:
 - υ COOPERAN.
 - υ SE SINCRONIZAN ENTRE SI.
- v EJEMPLOS:
 - υ FORMA DE IMPLANTAR LAS REGIONES CRITICAS.
 - υ FORMA DE ASIGNAR RECURSOS EN UN SISTEMA DISTRIBUIDO.
- v LOS PROBLEMAS RELATIVOS A LAS REGIONES CRITICAS, EXCLUSION MUTUA Y LA SINCRONIZACION:
 - υ GENERALMENTE SE RESUELVEN EN SISTEMAS DE UNA SOLA CPU CON METODOS COMO LOS SEMAFOROS Y LOS MONITORES:
 - Φ SE BASAN EN LA MEMORIA COMPARTIDA:
 - NO SON APLICABLES A SISTEMAS DISTRIBUIDOS.
- v OTRO PROBLEMA DE GRAN IMPORTANCIA ES EL TIEMPO Y LA FORMA DE MEDIRLO:
 - υ JUEGA UN PAPEL FUNDAMENTAL EN ALGUNOS MODELOS DE SINCRONIZACION.

SINCRONIZACION DE RELOJES

- v GENERALMENTE LOS ALGORITMOS DISTRIBUIDOS TIENEN LAS SIGUIENTES PROPIEDADES:
 - υ LA INFORMACION RELEVANTE SE DISTRIBUYE ENTRE VARIAS MAQUINAS.
 - υ LOS PROCESOS TOMAN LAS DECISIONES SOLO CON BASE EN LA INFORMACION DISPONIBLE EN FORMA LOCAL.
 - υ DEBE EVITARSE UN UNICO PUNTO DE FALLO EN EL SISTEMA.
 - υ NO EXISTE UN RELOJ COMUN O ALGUNA OTRA FUENTE PRECISA DEL TIEMPO GLOBAL.
- v LOS PRIMEROS TRES PUNTOS INDICAN QUE ES INACEPTABLE REUNIR TODA LA INFORMACION EN UN SOLO LUGAR PARA SU PROCESAMIENTO:
 - υ LOGRAR LA SINCRONIZACION SIN CENTRALIZACION REQUIERE HACER LAS COSAS DISTINTAS AL CASO DE LOS SISTEMAS OPERATIVOS TRADICIONALES.
- v EL ULTIMO PUNTO TAMBIEN ES CRUCIAL:
 - υ EN UN SISTEMA CENTRALIZADO EL TIEMPO NO ES AMBIGUO.
 - υ EN UN SISTEMA DISTRIBUIDO NO ES TRIVIAL PONER DE ACUERDO A TODAS LAS MAQUINAS EN LA HORA.
 - υ SE REQUIERE UN ACUERDO GLOBAL EN EL TIEMPO:
 - ⊕ LA FALTA DE SINCRONIZACION EN LOS RELOJES PUEDE SER DRASTICA EN PROCESOS DEPENDIENTES DEL TIEMPO.
- v LA PREGUNTA ES SI ES POSIBLE SINCRONIZAR TODOS LOS RELOJES EN UN SISTEMA DISTRIBUIDO.

- v LAS COMPUTADORAS POSEEN UN CIRCUITO PARA EL REGISTRO DEL TIEMPO CONOCIDO COMO **DISPOSITIVO RELOJ**.
- v ES UN **CRONOMETRO** CONSISTENTE EN UN CRISTAL DE CUARZO DE PRECISION SOMETIDO A UNA TENSION ELECTRICA QUE:
 - u OSCILA CON UNA FRECUENCIA BIEN DEFINIDA QUE DEPENDE DE:
 - Φ AL FORMA EN QUE SE CORTE EL CRISTAL.
 - Φ EL TIPO DE CRISTAL.
 - Φ LA MAGNITUD DE LA TENSION.
 - u A CADA CRISTAL SE LE ASOCIAN DOS REGISTROS:
 - Φ **R. CONTADOR.**
 - Φ **R. MANTENEDOR.**
 - u CADA OSCILACION DEL CRISTAL DECREMENTA EN 1 AL CONTADOR.
 - u CUANDO EL CONTADOR LLEGA A "0":
 - Φ SE GENERA UNA INTERRUPCION.
 - Φ EL CONTADOR SE VUELVE A CARGAR MEDIANTE EL R. MANTENEDOR.
 - u SE PUEDE PROGRAMAR UN CRONOMETRO PARA QUE GENERE UNA INTERRUPCION "X" VECES POR SEGUNDO.
 - u CADA INTERRUPCION SE DENOMINA **MARCA DE RELOJ**.
- v PARA UNA COMPUTADORA Y UN RELOJ:
 - u NO INTERESAN PEQUEÑOS DESFASAJES DEL RELOJ PORQUE:
 - Φ TODOS LOS PROCESOS DE LA MAQUINA USAN EL MISMO RELOJ Y TENDRAN CONSISTENCIA INTERNA.
 - Φ IMPORTAN LOS TIEMPOS RELATIVOS.
- v PARA VARIAS COMPUTADORAS CON SUS RESPECTIVOS RELOJES:
 - u ES IMPOSIBLE GARANTIZAR QUE LOS CRISTALES DE COMPUTADORAS DISTINTAS OSCILEN CON LA MISMA FRECUENCIA.
 - u HABRA UNA PERDIDA DE SINCRONIA EN LOS RELOJES (DE SOFTWARE):
 - Φ TENDRAN VALORES DISTINTOS AL SER LEIDOS.

- v LA DIFERENCIA ENTRE LOS VALORES DEL TIEMPO SE LLAMA **DISTORSION DEL RELOJ**:
 - v PODRIA GENERAR FALLAS EN LOS PROGRAMAS DEPENDIENTES DEL TIEMPO.
- v LAMPORT DEMOSTRO QUE LA SINCRONIZACION DE RELOJES ES POSIBLE Y PRESENTO UN ALGORITMO PARA LOGRARLO.
- v LAMPORT SEÑALO QUE LA SINCRONIZACION DE RELOJES NO TIENE QUE SER ABSOLUTA:
 - v SI 2 PROCESOS NO INTERACTUAN NO ES NECESARIO QUE SUS RELOJES ESTEN SINCRONIZADOS.
 - v GENERALMENTE LO IMPORTANTE NO ES QUE LOS PROCESOS ESTEN DE ACUERDO EN LA HORA:
 - Φ SI IMPORTA QUE COINCIDAN EN EL ORDEN EN QUE OCURREN LOS EVENTOS.
- v PARA CIERTOS ALGORITMOS LO QUE IMPORTA ES LA CONSISTENCIA INTERNA DE LOS RELOJES:
 - v NO INTERESA SU CERCANIA PARTICULAR AL TIEMPO REAL (OFICIAL).
 - v LOS RELOJES SE DENOMINAN **RELOJES LOGICOS**.
- v LOS **RELOJES FISICOS** SON RELOJES QUE:
 - v DEBEN SER IGUALES (ESTAR SINCRONIZADOS).
 - v NO DEBEN DESVIARSE DEL TIEMPO REAL MAS ALLA DE CIERTA MAGNITUD.
- v PARA SINCRONIZAR LOS RELOJES LOGICOS, **LAMPORT** DEFINIO LA RELACION **OCURRE ANTES DE (HAPPENS-BEFORE)**:
 - v SI a Y b SON EVENTOS EN EL MISMO PROCESO Y a OCURRE ANTES DE b , ENTONCES $a \rightarrow b$ ES VERDADERO.
 - v “OCURRE ANTES DE” ES UNA RELACION TRANSITIVA:
 - Φ SI $a \rightarrow b$ Y $b \rightarrow c$, ENTONCES $a \rightarrow c$.
 - v SI DOS EVENTOS x E y ESTAN EN PROCESOS DIFERENTES QUE NO INTERCAMBIAN MENSAJES, ENTONCES $x \rightarrow y$ NO ES VERDADERO, PERO TAMPOCO LO ES $y \rightarrow x$:
 - Φ SE DICE QUE SON **EVENTOS CONCURRENTES**.

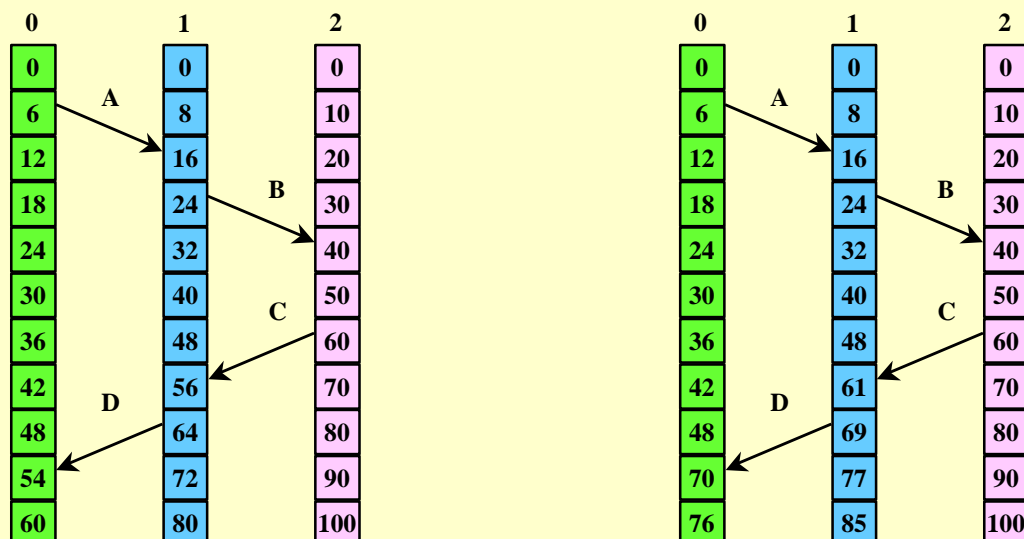
RELOJES LOGICOS

- v NECESITAMOS UNA FORMA DE MEDIR EL TIEMPO TAL QUE A CADA EVENTO a , LE PODAMOS ASOCIAR UN VALOR DEL TIEMPO $C(a)$ EN EL QUE TODOS LOS PROCESOS ESTEN DE ACUERDO:
 - v SE DEBE CUMPLIR QUE:
 - ⊕ SI $a \rightarrow b$ ENTONCES $C(a) < C(b)$.
 - ⊕ EL TIEMPO DEL RELOJ, C , SIEMPRE DEBE IR HACIA ADELANTE (CRECIENTE), Y NUNCA HACIA ATRAS (DECRECIENTE).
- v EL ALGORITMO DE LAMPORT ASIGNA TIEMPOS A LOS EVENTOS.
- v CONSIDERAMOS TRES PROCESOS QUE SE EJECUTAN EN DIFERENTES MAQUINAS, C/U CON SU PROPIO RELOJ Y VELOCIDAD:
 - v EL PROCESO "0" ENVIA EL MENSAJE "A" AL PROCESO "1" CUANDO EL RELOJ DE "0" MARCA "6".
 - v EL PROCESO "1" RECIBE EL MENSAJE "A" CUANDO SU RELOJ MARCA "16".
 - v SI EL MENSAJE ACARREA EL TIEMPO DE INICIO "6", EL PROCESO "1" CONSIDERARA QUE TARDO 10 MARCAS DE RELOJ EN VIAJAR.
 - v EL MENSAJE "B" DE "1" A "2" TARDA 16 MARCAS DE RELOJ.
 - v EL MENSAJE "C" DE "2" A "1" SALE EN "60" Y LLEGA EN "56", TARDARIA UN TIEMPO NEGATIVO, LO CUAL ES IMPOSIBLE.
 - v EL MENSAJE "D" DE "1" A "0" SALE EN "64" Y LLEGA EN "54".
 - v LAMPORT UTILIZA LA RELACION "OCURRE ANTES DE":
 - ⊕ SI "C" SALE EN "60" DEBE LLEGAR EN "61" O EN UN TIEMPO POSTERIOR.
 - ⊕ CADA MENSAJE ACARREA EL TIEMPO DE ENVIO, DE ACUERDO CON EL RELOJ DEL EMISOR.
 - ⊕ CUANDO UN MENSAJE LLEGA Y EL RELOJ DEL RECEPTOR MUESTRA UN VALOR ANTERIOR AL TIEMPO EN QUE SE ENVIO EL MENSAJE:
 - EL RECEPTOR ADELANTA SU RELOJ PARA QUE TENGA UNA UNIDAD MAS QUE EL TIEMPO DE ENVIO.

RELOJES LOGICOS

- v ESTE ALGORITMO CUMPLE NUESTRAS NECESIDADES PARA EL TIEMPO GLOBAL, SI SE HACE EL SIGUIENTE AGREGADO:
 - v ENTRE DOS EVENTOS CUALESQUIERA, EL RELOJ DEBE MARCAR AL MENOS UNA VEZ.
 - v DOS EVENTOS NO DEBEN OCURRIR EXACTAMENTE AL MISMO TIEMPO.
- v CON ESTE ALGORITMO SE PUEDE ASIGNAR UN TIEMPO A TODOS LOS EVENTOS EN UN SISTEMA DISTRIBUIDO, CON LAS SIGUIENTES CONDICIONES:
 - v SI a OCURRE ANTES DE b EN EL MISMO PROCESO, $C(a) < C(b)$.
 - v SI a Y b SON EL ENVIO Y RECEPCION DE UN MENSAJE, $C(a) < C(b)$.
 - v PARA TODOS LOS EVENTOS a Y b , $C(a)$ ES DISTINTO DE $C(b)$.

EJEMPLO DE TRES PROCESOS CUYOS RELOJES CORREN A DIFERENTES VELOCIDADES - EL ALGORITMO DE LAMPORT CORRIGE LOS RELOJES



RELOJES FISICOS

- v EL ALGORITMO DE LAMPORT PROPORCIONA UN ORDEN DE EVENTOS SIN AMBIGÜEDADES, PERO:
 - v LOS VALORES DE TIEMPO ASIGNADOS A LOS EVENTOS NO TIENEN PORQUE SER CERCANOS A LOS TIEMPOS REALES EN LOS QUE OCURREN.
 - v EN CIERTOS SISTEMAS (EJ.: SIST. DE TIEMPO REAL), ES IMPORTANTE LA HORA REAL DEL RELOJ:
 - Φ SE PRECISAN RELOJES FISICOS EXTERNOS (MAS DE UNO).
 - Φ SE DEBEN SINCRONIZAR:
 - CON LOS RELOJES DEL MUNDO REAL.
 - ENTRE SI.
- v LA MEDICION DEL TIEMPO REAL CON ALTA PRECISION NO ES SENCILLA.
- v DESDE ANTIGUO EL TIEMPO SE HA MEDIDO ASTRONOMICAMENTE.
- v SE CONSIDERA EL **DIA SOLAR** AL INTERVALO ENTRE DOS TRANSITOS CONSECUTIVOS DEL SOL:
 - v EL **TRANSITO DEL SOL** ES EL EVENTO EN QUE EL SOL ALCANZA SU PUNTO APARENTEMENTE MAS ALTO EN EL CIELO.
- v EL **SEGUNDO SOLAR** SE DEFINE COMO 1 / 86400 DE UN DIA SOLAR.
- v COMO EL PERIODO DE ROTACION DE LA TIERRA NO ES CONSTANTE, SE CONSIDERA EL **SEGUNDO SOLAR PROMEDIO** DE UN GRAN N° DE DIAS.
- v LOS FISICOS DEFINIERON AL **SEGUNDO** COMO EL TIEMPO QUE TARDA EL ATOMO DE CESIO 133 PARA HACER 9.192.631.770 TRANSICIONES
 - v SE TOMO ESTE N° PARA QUE EL **SEGUNDO ATOMICO** COINCIDA CON EL **SEGUNDO SOLAR PROMEDIO** DE 1958.
- v LA OFICINA INTERNACIONAL DE LA HORA EN PARIS (BIH) RECIBE LAS INDICACIONES DE CERCA DE 50 RELOJES ATOMICOS EN EL MUNDO Y CALCULA EL **TIEMPO ATOMICO INTERNACIONAL (TAI)**.

RELOJES FISICOS

- v COMO CONSECUENCIA DE QUE EL **DIA SOLAR PROMEDIO** ES CADA VEZ MAYOR, UN **DIA TAI** ES 3 MSEG MENOR QUE UN **DSP**:
 - υ LA BIH INTRODUCE **SEGUNDOS DE SALTO** PARA HACER LAS CORRECCIONES NECESARIAS PARA QUE PERMANEZCAN EN FASE:
 - Φ EL SISTEMA DE TIEMPO BASADO EN LOS SEGUNDOS TAI.
 - Φ EL MOVIMIENTO APARENTE DEL SOL.
 - υ SURGE EL **TIEMPO COORDENADO UNIVERSAL (UTC)**.
- v EL INSTITUTO NACIONAL DEL TIEMPO ESTANDAR (NIST) DE EE.UU. Y OTROS DE OTROS PAISES:
 - υ OPERAN ESTACIONES DE RADIO DE ONDA CORTA O SATELITES DE COMUNICACIONES.
 - υ TRANSMITEN PULSOS UTC CON CIERTA REGULARIDAD ESTABLECIDA (CADA SEGUNDO, CADA 0,5 MSEG, ETC.).
 - υ SE DEBEN CONOCER CON PRECISION LA POSICION RELATIVA DEL EMISOR Y DEL RECEPTOR:
 - Φ SE DEBE COMPENSAR EL RETRASO DE PROPAGACION DE LA SEÑAL.
 - Φ SI LA SEÑAL SE RECIBE POR MODEM TAMBIEN SE DEBE COMPENSAR POR LA RUTA DE LA SEÑAL Y LA VELOCIDAD DEL MODEM.
 - Φ SE DIFICULTA LA OBTENCION DEL TIEMPO CON UNA PRECISION EXTREMADAMENTE ALTA.

ALGORITMOS PARA LA SINCRONIZACION DE RELOJES

- v SI UNA MAQUINA TIENE UN RECEPTOR DE UTC:
 - u TODAS LAS MAQUINAS DEBEN SINCRONIZARSE CON ELLA.
- v SI NINGUNA MAQUINA TIENE UN RECEPTOR DE UTC:
 - u CADA MAQUINA LLEVA EL REGISTRO DE SU PROPIO TIEMPO.
 - u SE DEBE MANTENER EL TIEMPO DE TODAS LAS MAQUINAS TAN CERCANO COMO SEA POSIBLE.
- v SE SUPONE QUE CADA MAQUINA TIENE UN CRONOMETRO QUE PROVOCA UNA INTERRUPCION “H” VECES POR SEGUNDO.
- v CUANDO EL CRONOMETRO SE DETIENE, EL MANEJADOR DE INTERRUPCIONES AÑADE “1” A UN RELOJ EN SOFTWARE.
- v EL RELOJ EN SOFTWARE MANTIENE UN REGISTRO DEL N° DE MARCAS (INTERRUPCIONES) A PARTIR DE CIERTA FECHA ACORDADA ANTES:
 - u AL VALOR DE ESTE RELOJ SE LO LLAMA “C”.
- v CUANDO EL TIEMPO UTC ES “t”, EL VALOR DEL RELOJ EN LA MAQUINA “p” ES “Cp(t)”:
 - u LO IDEAL SERIA QUE “Cp(t)” = “t” PARA TODA “p” Y TODA “t”:
 - Φ “dC/dt” DEBERIA SER “1”.
 - u LO REAL ES QUE LOS CRONOMETROS NO REALIZAN INTERRUPCIONES EXACTAMENTE “H” VECES POR SEGUNDO:
 - Φ POSEEN UN ERROR RELATIVO DE APROXIMADAMENTE 10^{-5} .
 - Φ EL FABRICANTE ESPECIFICA UNA CONSTANTE ρ LLAMADA **TASA MAXIMA DE ALEJAMIENTO** QUE ACOTA EL ERROR.
 - Φ EL CRONOMETRO FUNCIONA DENTRO DE SU ESPECIFICACION SI EXISTE UNA CONSTANTE ρ Y SE CUMPLE:
 - $1 - \rho \leq dC / \delta\tau \leq 1 + \rho$.
- v SI DOS RELOJES SE ALEJAN DE UTC EN LA DIRECCION OPUESTA:
 - u EN UN INSTANTE Δt LUEGO DE LA SINCRONIZACION PODRIAN ESTAR TAN ALEJADOS COMO: $2 \rho \Delta t$.
 - u PARA GARANTIZAR QUE NO DIFIERAN MAS DE δ :
 - Φ SE DEBEN VOLVER A SINCRONIZAR (EN SOFTWARE) AL MENOS CADA $\delta / 2 \rho$ SEGUNDOS.

ALGORITMOS PARA LA SINCRONIZACION DE RELOJES

- v **ALGORITMO DE CRISTIAN.**
- v ES ADECUADO PARA SISTEMAS EN LOS QUE:
 - v UNA MAQUINA TIENE UN RECEPTOR UTC:
 - Φ SE LA LLAMA **DESPACHADOR DEL TIEMPO**.
 - v EL OBJETIVO ES SINCRONIZAR TODAS LAS MAQUINAS CON ELLA.
- v CADA MAQUINA ENVIA UN MENSAJE AL SERVIDOR PARA SOLICITAR EL TIEMPO ACTUAL:
 - v PERIODICAMENTE, EN UN TIEMPO NO MAYOR QUE $\delta / 2$ p SEGUNDOS.
- v EL DESPACHADOR DEL TIEMPO RESPONDE PRONTAMENTE CON UN MENSAJE QUE CONTIENE EL TIEMPO ACTUAL C_{UTC} .
- v CUANDO EL EMISOR OBTIENE LA RESPUESTA PUEDE HACER QUE SU TIEMPO SEA C_{UTC} .
- v UN GRAN PROBLEMA ES QUE EL TIEMPO NO PUEDE CORRER HACIA ATRAS:
 - v C_{UTC} NO PUEDE SER MENOR QUE EL TIEMPO ACTUAL “C” DEL EMISOR.
 - v LA ATENCION DEL REQUERIMIENTO EN EL **SERVIDOR DE TIEMPOS** REQUIERE UN TIEMPO DEL MANEJADOR DE INTERRUPCIONES.
 - v TAMBIEN SE DEBE CONSIDERAR EL TIEMPO DE TRANSMISION.
- v EL CAMBIO DEL RELOJ SE DEBE INTRODUCIR DE MANERA GLOBAL:
 - v SI EL CRONOMETRO GENERA 100 INTERRUPCIONES POR SEGUNDO:
 - Φ CADA INTERRUPCION AÑADE 10 MSEG AL TIEMPO.
 - Φ PARA ATRASAR SOLO AGREGARIA 9 MSEG.
 - Φ PARA ADELANTAR AGREGARIA 11 MSEG.
- v LA CORRECCION POR EL TIEMPO DEL SERVIDOR Y EL TIEMPO DE TRANSMISION SE HACE MIDIENDO EN EL EMISOR:
 - v EL TIEMPO INICIAL (ENVIO) T_0 .
 - v EL TIEMPO FINAL (RECEPCION) T_1 .
 - v AMBOS TIEMPOS SE MIDEN CON EL MISMO RELOJ.
- v EL *TIEMPO DE PROPAGACION* DEL MENSAJE SERA $(T_1 - T_0) / 2$.

ALGORITMOS PARA LA SINCRONIZACION DE RELOJES

- v SI EL TIEMPO DEL SERVIDOR PARA MANEJAR LA INTERRUPCION Y PROCESAR EL MENSAJE ES “T”:
 - υ EL *TIEMPO DE PROPAGACION* SERA $(T_1 - T_0 - I) / 2$.
- v PARA MEJORAR LA PRECISION:
 - υ SE TOMAN VARIAS MEDICIONES.
 - υ SE DESCARTAN LOS VALORES EXTREMOS.
 - υ SE PROMEDIA EL RESTO.
- v EL TIEMPO DE PROPAGACION SE SUMA AL TIEMPO DEL SERVIDOR PARA SINCRONIZAR AL EMISOR CUANDO ESTE RECIBE LA RESPUESTA.
- v **ALGORITMO DE BERKELEY.**
- v EN EL ALGORITMO DE CRISTIAN EL SERVIDOR DE TIEMPO ES PASIVO.
- v EN EL ALGORITMO DE BERKELEY EL SERVIDOR DE TIEMPO:
 - υ ES ACTIVO.
 - υ REALIZA UN MUESTREO PERIODICO DE TODAS LAS MAQUINAS PARA PREGUNTARLES EL TIEMPO.
 - υ CON LAS RESPUESTAS:
 - Φ CALCULA UN TIEMPO PROMEDIO.
 - Φ INDICA A LAS DEMAS MAQUINAS QUE AVANCEN SU RELOJ O DISMINUYAN LA VELOCIDAD DEL MISMO HASTA LOGRAR LA DISMINUCION REQUERIDA.
- v ES ADECUADO CUANDO NO SE DISPONE DE UN RECEPTOR UTC.
- v **ALGORITMOS CON PROMEDIO.**
- v LOS ANTERIORES SON ALGORITMOS CENTRALIZADOS.
- v UNA CLASE DE ALGORITMOS DESCENTRALIZADOS DIVIDE EL TIEMPO EN INTERVALOS DE RESINCRONIZACION DE LONGITUD FIJA:
 - υ EL *i*-ESIMO INTERVALO:
 - Φ INICIA EN “ $T_0 + iR$ ” Y VA HASTA “ $T_0 + (i + 1) R$ ”.
 - Φ “ T_0 ” ES UN MOMENTO YA ACORDADO EN EL PASADO.
 - Φ “ R ” ES UN PARAMETRO DEL SISTEMA.
- v AL INICIO DE CADA INTERVALO CADA MAQUINA TRANSMITE EL TIEMPO ACTUAL SEGUN SU RELOJ.
- v DEBIDO A LA DIFERENTE VELOCIDAD DE LOS RELOJES LAS TRANSMISIONES NO SERAN SIMULTANEAS.

ALGORITMOS PARA LA SINCRONIZACION DE RELOJES

- v LUEGO DE QUE UNA MAQUINA TRANSMITE SU HORA:
 - INICIALIZA UN CRONOMETRO LOCAL PARA REUNIR LAS DEMAS TRANSMISIONES QUE LLEGUEN EN CIERTO INTERVALO “S”.
- v CUANDO RECIBE TODAS LAS TRANSMISIONES SE EJECUTA UN ALGORITMO PARA CALCULAR UNA NUEVA HORA PARA LOS RELOJES.
- v UNA VARIANTE ES PROMEDIAR LOS VALORES DE TODAS LAS DEMAS MAQUINAS.
- v OTRA VARIANTE ES DESCARTAR LOS VALORES EXTREMOS ANTES DE PROMEDIAR (LOS “m” MAYORES Y LOS “m” MENORES).
- v UNA MEJORA AL ALGORITMO CONSIDERA LA CORRECCION POR TIEMPOS DE PROPAGACION.
- v **VIAS FUENTES EXTERNAS DE TIEMPO.**
- v LOS SISTEMAS QUE REQUIEREN UNA SINCRONIZACION MUY PRECISA CON UTC SE PUEDEN EQUIPAR CON VARIOS RECEPTORES DE UTC.
- v LAS DISTINTAS FUENTES DE TIEMPO GENERARAN DISTINTOS RANGOS (INTERVALOS DE TIEMPO) DONDE “CAERAN” LOS RESPECTIVOS UTC:
 - ES NECESARIA UNA SINCRONIZACION.
- v COMO LA TRANSMISION NO ES INSTANTANEA SE GENERA UNA CIERTA INCERTIDUMBRE EN EL TIEMPO.
- v CUANDO UN PROCESADOR OBTIENE TODOS LOS RANGOS DE UTC:
 - VERIFICA SI ALGUNO DE ELLOS ES AJENO A LOS DEMAS Y DE SERLO LO DESCARTA POR SER UN VALOR EXTREMO.
 - CALCULA LA INTERSECCION (EN EL TIEMPO) DE LOS DEMAS RANGOS.
 - LA INTERSECCION DETERMINA UN INTERVALO CUYO PUNTO MEDIO SERA EL UTC Y LA HORA DEL RELOJ INTERNO.
- v SE DEBEN COMPENSAR LOS RETRASOS DE TRANSMISION Y LAS DIFERENCIAS DE VELOCIDADES DE LOS RELOJES.
- v SE DEBE ASEGURAR QUE EL TIEMPO NO CORRA HACIA ATRAS.
- v SE DEBE RESINCRONIZAR PERIODICAMENTE DESDE LAS FUENTES EXTERNAS DE UTC.

EXCLUSION MUTUA

- v CUANDO UN PROCESO DEBE LEER O ACTUALIZAR CIERTAS ESTRUCTURAS DE DATOS COMPARTIDAS:
 - v PRIMERO INGRESA A UNA REGION CRITICA PARA:
 - ⊕ LOGRAR LA EXCLUSION MUTUA Y GARANTIZAR QUE NINGUN OTRO PROCESO UTILIZARA LAS ESTRUCTURAS DE DATOS AL MISMO TIEMPO.
- v EN SISTEMAS MONOPROCESADORES LAS REGIONES CRITICAS SE PROTEGEN CON SEMAFOROS, MONITORES Y SIMILARES.
- v EN SISTEMAS DISTRIBUIDOS LA CUESTION ES MAS COMPLEJA.
- v **UN ALGORITMO CENTRALIZADO.**
- v LA FORMA MAS DIRECTA DE LOGRAR LA EXCLUSION MUTUA EN UN SISTEMA DISTRIBUIDO ES SIMULAR A LA FORMA EN QUE SE LLEVA A CABO EN UN SISTEMA MONOPROCESADOR.
- v SE ELIGE UN *PROCESO COORDINADOR*.
- v CUANDO UN PROCESO DESEA INGRESAR A UNA REGION CRITICA:
 - v ENVIA UN MENSAJE DE SOLICITUD AL COORDINADOR:
 - ⊕ INDICANDO LA REGION CRITICA.
 - ⊕ SOLICITANDO PERMISO DE ACCESO.
 - v SI NINGUN OTRO PROCESO ESTA EN ESE MOMENTO EN ESA REGION CRITICA:
 - ⊕ EL COORDINADOR ENVIA UNA RESPUESTA OTORGANDO EL PERMISO.
 - v CUANDO LLEGA LA RESPUESTA EL PROCESO SOLICITANTE ENTRA A LA REGION CRITICA.
- v SI UN PROCESO PIDE PERMISO PARA ENTRAR A UNA REGION CRITICA YA ASIGNADA A OTRO PROCESO:
 - v EL COORDINADOR NO OTORGA EL PERMISO Y ENCOLA EL PEDIDO.
- v CUANDO UN PROCESO SALE DE LA REGION CRITICA ENVIA UN MENSAJE AL COORDINADOR PARA LIBERAR SU ACCESO EXCLUSIVO:
 - v EL COORDINADOR EXTRAE EL PRIMER ELEMENTO DE LA COLA DE SOLICITUDES DIFERIDAS Y ENVIA A ESE PROCESO UN MENSAJE OTORGANDO EL PERMISO:
 - ⊕ EL PROCESO QUEDA HABILITADO PARA ACCEDER A LA REGION CRITICA SOLICITADA.
- v ES UN ESQUEMA SENCILLO, JUSTO Y CON POCOS MENSAJES DE CONTROL.
- v LA LIMITANTE ES QUE EL COORDINADOR PUEDE SER UN CUELLO DE BOTELLA Y PUEDE FALLAR Y BLOQUEAR A LOS PROCESOS QUE ESPERAN UNA RESPUESTA DE HABILITACION DE ACCESO.

EXCLUSION MUTUA

- v **UN ALGORITMO DISTRIBUIDO.**
- v EL OBJETIVO ES NO TENER UN UNICO PUNTO DE FALLO (EL COORDINADOR CENTRAL).
- v UN EJ. ES EL ALGORITMO DE LAMPORT MEJORADO POR RICART Y AGRAWALA.
- v SE REQUIERE UN ORDEN TOTAL DE TODOS LOS EVENTOS EN EL SISTEMA PARA SABER CUAL OCURRIO PRIMERO.
- v CUANDO UN PROCESO DESEA ENTRAR A UNA REGION CRITICA:
 - u CONSTRUYE UN MENSAJE CON EL NOMBRE DE LA REGION CRITICA, SU N° DE PROCESO Y LA HORA ACTUAL.
 - u ENVIA EL MENSAJE A TODOS LOS DEMAS PROCESOS Y DE MANERA CONCEPTUAL A EL MISMO.
 - u SE SUPONE QUE CADA MENSAJE TIENE UN RECONOCIMIENTO.
- v SI EL RECEPTOR NO ESTA EN LA REGION CRITICA Y NO DESEA ENTRAR A ELLA, ENVIA DE REGRESO UN MENSAJE OK AL EMISOR.
- v SI EL RECEPTOR YA ESTA EN LA REGION CRITICA NO RESPONDE Y ENCOLA LA SOLICITUD.
- v SI EL RECEPTOR DESEA ENTRAR A LA REGION CRITICA PERO AUN NO LO LOGRO, COMPARA:
 - u LA MARCA DE TIEMPO DEL MENSAJE RECIBIDO CON.
 - u LA MARCA CONTENIDA EN EL MENSAJE QUE ENVIO A CADA UNO.
 - u LA MENOR DE LAS MARCAS GANA.
 - u SI EL MENSAJE RECIBIDO ES MENOR EL RECEPTOR ENVIA UN OK.
 - u SI SU PROPIO MENSAJE TIENE UNA MARCA MENOR EL RECEPTOR NO ENVIA NADA Y ENCOLA EL PEDIDO.
- v LUEGO DE ENVIAR LAS SOLICITUDES UN PROCESO:
 - u ESPERA HASTA QUE ALGUIEN MAS OBTIENE EL PERMISO.
 - u CUANDO LLEGAN TODOS LOS PERMISOS PUEDE ENTRAR A LA REGION CRITICA.
- v CUANDO UN PROCESO SALE DE LA REGION CRITICA:
 - u ENVIA MENSAJES OK A TODOS LOS PROCESOS EN SU COLA.
 - u ELIMINA A TODOS LOS ELEMENTOS DE LA COLA.

EXCLUSION MUTUA

- v LA EXCLUSION MUTUA QUEDA GARANTIZADA SIN BLOQUEO NI INANICION.
- v EL N° DE MENSAJES NECESARIOS POR ENTRADA ES " $2(n - 1)$ ", SIENDO " n " EL N° TOTAL DE PROCESOS EN EL SISTEMA.
- v NO EXISTE UN UNICO PUNTO DE FALLO SINO " n ":
 - u SI CUALQUIER PROCESO FALLA NO RESPONDERA A LAS SOLICITUDES.
 - u LA FALTA DE RESPUESTA SE INTERPRETARA COMO NEGACION DE ACCESO:
 - Φ SE BLOQUEARAN LOS SIGUIENTES INTENTOS DE LOS DEMAS PROCESOS POR ENTRAR A TODAS LAS REGIONES CRITICAS.
- v SE INCREMENTA LA PROBABILIDAD DE FALLO EN " n " VECES Y TAMBIEN EL TRAFICO EN LA RED.
- v SE PUEDE SOLUCIONAR EL BLOQUEO SI:
 - u EL EMISOR ESPERA Y SIGUE INTENTANDO HASTA QUE REGRESA UNA RESPUESTA O.
 - u EL EMISOR CONCLUYE QUE EL DESTINATARIO ESTA FUERA DE SERVICIO.
- v OTRO PROBLEMA ES QUE:
 - u SE UTILIZARA UNA PRIMITIVA DE COMUNICACION EN GRUPO O.
 - u CADA PROCESO DEBE MANTENER LA LISTA DE MIEMBROS DEL GRUPO, INCLUYENDO LOS PROCESOS QUE INGRESAN, LOS QUE SALEN Y LOS QUE FALLAN.
 - u SE COMPLICA PARA GRAN N° DE PROCESOS.
- v UN IMPORTANTE PROBLEMA ADICIONAL ES QUE:
 - u *TODOS* LOS PROCESOS PARTICIPAN EN *TODAS* LAS DECISIONES REFERENTES A LAS ENTRADAS EN LAS REGIONES CRITICAS.
 - u SE SOBRECARGA EL SISTEMA.
- v UNA MEJORA CONSISTE EN PERMITIR QUE UN PROCESO ENTRE A UNA REGION CRITICA CON EL PERMISO DE UNA MAYORIA SIMPLE DE LOS DEMAS PROCESOS (EN VEZ DE TODOS):
 - u LUEGO DE QUE UN PROCESO OTORGO EL PERMISO A OTRO PARA ENTRAR A UNA REGION CRITICA:
 - Φ NO PUEDE OTORGAR EL MISMO PERMISO A OTRO PROCESO HASTA QUE EL PRIMERO LIBERE SU PERMISO.

EXCLUSION MUTUA

- v **UN ALGORITMO DE ANILLO DE FICHAS (TOKEN RING).**
- v LOS PROCESOS SE ORGANIZAN POR SOFTWARE FORMANDO UN ANILLO LOGICO ASIGNANDOSE A CADA PROCESO UNA POSICION EN EL ANILLO.
- v CADA PROCESO SABE CUAL ES EL SIGUIENTE LUEGO DE EL.
- v AL INICIALIZAR EL ANILLO SE LE DA AL PROCESO "0" UNA **FICHA (TOKEN)** QUE CIRCULA EN TODO EL ANILLO:
 - v SE TRANSFIERE DEL PROCESO "k" AL "k + 1" EN MENSAJES PUNTUALES.
- v CUANDO UN PROCESO OBTIENE LA FICHA DE SU VECINO VERIFICA SI INTENTA ENTRAR A UNA REGION CRITICA:
 - v EN CASO POSITIVO:
 - Φ EL PROCESO ENTRA A LA REGION CRITICA, HACE EL PROCESO NECESARIO Y SALE DE ELLA.
 - Φ DESPUES DE SALIR PASA LA FICHA A LO LARGO DEL ANILLO:
 - NO SE PUEDE ENTRAR A UNA SEGUNDA REGION CRITICA CON LA MISMA FICHA (TOKEN O PERMISO).
 - v EN CASO NEGATIVO:
 - Φ LA VUELVE A PASAR.
- v EN UN INSTANTE DADO SOLO UN PROCESO PUEDE ESTAR EN UNA REGION CRITICA.
- v SI LA FICHA SE PIERDE DEBE SER REGENERADA, PERO ES DIFICIL DETECTAR SU PERDIDA:
 - v LA CANTIDAD DE TIEMPO ENTRE LAS APARICIONES SUCESIVAS DE LA FICHA EN LA RED NO ESTA ACOTADA:
 - Φ ES DIFICIL DECIDIR SI ESTA PERDIDA O DEMORADA EN ALGUN PROCESO QUE NO LA LIBERA.
- v LA FALLA DE UN PROCESO ES DETECTADA CUANDO SU VECINO INTENTA SIN EXITO PASARLE LA FICHA:
 - v SE LO DEBE ELIMINAR DEL GRUPO Y PASAR LA FICHA AL SIGUIENTE PROCESO ACTIVO:
 - Φ TODOS LOS PROCESOS DEBEN MANTENER LA CONFIGURACION ACTUAL DEL ANILLO.

ALGORITMOS DE ELECCION

- v SON LOS ALGORITMOS PARA LA ELECCION DE UN PROCESO COORDINADOR, INICIADOR, SECUENCIADOR, ETC.
- v EL **OBJETIVO** DE UN ALGORITMO DE ELECCION ES GARANTIZAR QUE INICIADA UNA ELECCION ESTA CONCLUYA CON EL ACUERDO DE TODOS LOS PROCESOS CON RESPECTO A LA IDENTIDAD DEL NUEVO COORDINADOR.
- v **EL ALGORITMO DEL GRANDULON O DE GARCIA-MOLINA.**
- v UN PROCESO “P” INICIA UNA ELECCION CUANDO OBSERVA QUE EL COORDINADOR YA NO RESPONDE A LAS SOLICITUDES.
- v “P” REALIZA UNA ELECCION DE LA SIGUIENTE MANERA:
 - v ENVIA UN MENSAJE *ELECCION* A LOS DEMAS PROCESOS CON UN N° MAYOR.
 - v SI NADIE RESPONDE ASUME QUE GANA LA ELECCION Y SE CONVIERTE EN EL NUEVO COORDINADOR.
 - v SI UN PROCESO CON UN N° MAYOR RESPONDE, TOMA EL CONTROL Y EL TRABAJO DE “P” TERMINA.
- v UN PROCESO PUEDE RECIBIR EN CUALQUIER MOMENTO UN MENSAJE *ELECCION* DE OTROS PROCESOS CON UN N° MENOR:
 - v ENVIA DE REGRESO UN MENSAJE *OK* AL EMISOR PARA INDICAR QUE ESTA VIVO Y QUE TOMARA EL CONTROL.
 - v REALIZA UNA ELECCION SALVO QUE YA ESTE HACIENDO ALGUNA.
- v EN CIERTO MOMENTO TODOS LOS PROCESOS HAN DECLINADO ANTE UNO DE ELLOS, QUE SERA EL NUEVO COORDINADOR:
 - v ENVIA UN MENSAJE *COORDINADOR* A TODOS LOS PROCESOS PARA ANUNCIARLO.
- v SI UN PROCESO INACTIVO SE ACTIVA REALIZA UNA ELECCION:
 - v SI EL TIENE EL N° MAS ALTO SERA EL NUEVO COORDINADOR:
 - ⊕ SIEMPRE GANA EL PROCESO QUE POSEE EL N° MAYOR, DE AHI EL NOMBRE “ALGORITMO DEL GRANDULON”.

ALGORITMOS DE ELECCION

- v **UN ALGORITMO DE ANILLO.**
- v SE SUPONE QUE LOS PROCESOS TIENEN UN ORDEN FISICO O LOGICO:
 - u CADA PROCESO CONOCE A SU SUCESOR.
- v CUANDO ALGUN PROCESO OBSERVA QUE EL COORDINADOR NO FUNCIONA:
 - u CONSTRUYE UN MENSAJE *ELECCION* CON SU PROPIO N° DE PROCESO.
 - u ENVIA EL MENSAJE A SU SUCESOR.
 - u SI EL SUCESOR ESTA INACTIVO:
 - Φ EL EMISOR VA HACIA EL SIGUIENTE N° DEL ANILLO O AL SIGUIENTE DE ESTE.
 - Φ CONTINUA HASTA LOCALIZAR UN PROCESO EN EJECUCION.
 - Φ EN CADA PASO, AL EMISOR AÑADE SU PROPIO N° DE PROCESO A LA LISTA EN EL MENSAJE.
 - u EN CIERTO MOMENTO EL MENSAJE REGRESA AL PROCESO QUE LO INICIO:
 - Φ EL PROCESO LO RECONOCE AL RECIBIR UN MENSAJE CON SU PROPIO N° DE PROCESO.
 - u EL MENSAJE DE *ELECCION* SE TRANSFORMA EN MENSAJE *COORDINADOR* Y CIRCULA NUEVAMENTE:
 - Φ INFORMA A LOS DEMAS PROCESOS:
 - QUIEN ES EL COORDINADOR:
 - EL MIEMBRO DE LA LISTA CON EL N° MAYOR.
 - QUIENES SON LOS MIEMBROS DEL NUEVO ANILLO.
 - u CONCLUIDA LA RONDA DE INFORMACION EL MENSAJE *COORDINADOR* SE ELIMINA Y CONTINUAN LOS PROCESOS.

TRANSACCIONES ATOMICAS

- v LAS TECNICAS DE SINCRONIZACION YA VISTAS SON DE BAJO NIVEL:
 - u EL PROGRAMADOR DEBE ENFRENTARSE DIRECTAMENTE CON LOS DETALLES DE:
 - Φ LA EXCLUSION MUTUA.
 - Φ EL MANEJO DE LAS REGIONES CRITICAS.
 - Φ AL PREVENCION DE BLOQUEOS.
 - Φ LA RECUPERACION DE FALLAS.
- v SE PRECISAN TECNICAS DE ABSTRACCION DE MAYOR NIVEL QUE:
 - u OCULTEN ESTOS ASPECTOS TECNICOS.
 - u PERMITAN A LOS PROGRAMADORES CONCENTRARSE EN LOS ALGORITMOS Y LA FORMA EN QUE LOS PROCESOS TRABAJAN JUNTOS EN PARALELO.
- v TAL ABSTRACCION LA LLAMAREMOS **TRANSACCION ATOMICA, TRANSACCION O ACCION ATOMICA.**
- v LA PRINCIPAL PROPIEDAD DE LA TRANSACCION ATOMICA ES EL “TODO O NADA”:
 - u O SE HACE TODO LO QUE SE TENIA QUE HACER COMO UNA UNIDAD O NO SE HACE NADA.
 - u EJEMPLO:
 - Φ UN CLIENTE LLAMA AL BANCO MEDIANTE UNA PC CON UN MODEM PARA:
 - RETIRAR DINERO DE UNA CUENTA.
 - DEPOSITAR EL DINERO EN OTRA CUENTA.
 - Φ LA OPERACION TIENE DOS ETAPAS.
 - Φ SI LA CONEXION TELEFONICA FALLA LUEGO DE LA PRIMER ETAPA PERO ANTE DE LA SEGUNDA:
 - HABRA UN RETIRO PERO NO UN DEPOSITO.
 - Φ LA SOLUCION CONSISTE EN AGRUPAR LAS DOS OPERACIONES EN UNA **TRANSACCION ATOMICA:**
 - LAS DOS OPERACIONES TERMINARIAN O NO TERMINARIA NINGUNA.
 - SE DEBE REGRESAR AL ESTADO INICIAL SI LA TRANSACCION NO PUEDE CONCLUIR.

EL MODELO DE TRANSACCION

- v SUPONDREMOS QUE:
 - EL SISTEMA CONSTA DE VARIOS PROCESOS INDEPENDIENTES QUE PUEDEN FALLAR ALEATORIAMENTE.
 - EL SOFTWARE SUBYACENTE MANEJA TRANSPARENTAMENTE LOS ERRORES DE COMUNICACION.
- v **ALMACENAMIENTO ESTABLE.**
- v SE PUEDE IMPLANTAR CON UNA PAREJA DE DISCOS COMUNES.
- v CADA BLOQUE DE LA UNIDAD 2 ES UNA COPIA EXACTA (ESPEJO) DEL BLOQUE CORRESPONDIENTE EN LA UNIDAD 1.
- v CUANDO SE ACTUALIZA UN BLOQUE:
 - PRIMERO SE ACTUALIZA Y VERIFICA EL BLOQUE DE LA UNIDAD 1.
 - LUEGO SE ACTUALIZA Y VERIFICA EL BLOQUE DE LA UNIDAD 2.
- v SI EL SISTEMA FALLA LUEGO DE ACTUALIZAR LA UNIDAD 1 Y ANTES DE ACTUALIZAR LA UNIDAD 2:
 - LUEGO DE LA RECUPERACION SE PUEDEN COMPARAR AMBOS DISCOS BLOQUE POR BLOQUE:
 - ⦿ SE PUEDE ACTUALIZAR LA UNIDAD 2 EN FUNCION DE LA 1.
- v SI SE DETECTA EL DETERIORO ESPONTANEO DE UN BLOQUE:
 - SE LO REGENERA PARTIENDO DEL BLOQUE CORRESPONDIENTE EN LA OTRA UNIDAD.
- v UN ESQUEMA DE ESTE TIPO ES ADECUADO PARA LAS APLICACIONES QUE REQUIEREN DE UN ALTO GRADO DE TOLERANCIA DE FALLOS:
 - EJ.: LAS TRANSACCIONES ATOMICAS.
- v **PRIMITIVAS DE TRANSACCION.**
- v DEBEN SER PROPORCIONADAS POR EL SISTEMA OPERATIVO O POR EL SISTEMA DE TIEMPO DE EJECUCION DEL LENGUAJE.
- v EJEMPLOS:
 - *BEGIN_TRANSACTION*: LOS COMANDOS SIGUIENTES FORMAN UNA TRANSACCION.
 - *END_TRANSACTION*: TERMINA LA TRANSACCION Y SE INTENTA UN COMPROMISO.
 - *ABORT_TRANSACTION*: SE ELIMINA LA TRANSACCION; SE RECUPERAN LOS VALORES ANTERIORES.

EL MODELO DE TRANSACCION

- v *READ*: SE LEEN DATOS DE UN ARCHIVO (O ALGUN OTRO OBJETO).
- v *WRITE*: SE ESCRIBEN DATOS EN UN ARCHIVO (O ALGUN OTRO OBJETO).
- v LAS OPERACIONES ENTRE *BEGIN* Y *END* FORMAN EL CUERPO DE LA TRANSACCION Y DEBEN EJECUTARSE TODAS O NINGUNA DE ELLAS:
 - v PUEDEN SER LLAMADAS AL SISTEMA, PROCEDIMIENTO DE BIBLIOTECA O ENUNCIADOS EN UN LENGUAJE.
- v **PROPIEDADES DE LAS TRANSACCIONES.**
- v LAS PROPIEDADES FUNDAMENTALES SON:
 - v *SERIALIZACION*:
 - Φ LAS TRANSACCIONES CONCURRENTES NO INTERFIEREN ENTRE SI.
 - v *ATOMICIDAD*:
 - Φ PARA EL MUNDO EXTERIOR, LA TRANSACCION OCURRE DE MANERA INDIVISIBLE.
 - v *PERMANENCIA*:
 - Φ UNA VEZ COMPROMETIDA UNA TRANSACCION, LOS CAMBIOS SON PERMANENTES.
- v LA *SERIALIZACION* GARANTIZA QUE SI DOS O MAS TRANSACCIONES SE EJECUTAN AL MISMO TIEMPO:
 - v EL RESULTADO FINAL APARECE COMO SI TODAS LAS TRANSACCIONES SE EJECUTASEN DE MANERA SECUENCIAL EN CIERTO ORDEN:
 - Φ PARA C / U DE ELLAS Y PARA LOS DEMAS PROCESOS.
- v LA *ATOMICIDAD* GARANTIZA QUE CADA TRANSACCION NO OCURRE O BIEN SE REALIZA EN SU TOTALIDAD:
 - v SE PRESENTA COMO UNA ACCION INDIVISIBLE E INSTANTANEA.
- v LA *PERMANENCIA* SE REFIERE A QUE UNA VEZ COMPROMETIDA UNA TRANSACCION:
 - v SIGUE ADELANTE Y LOS RESULTADOS SON PERMANENTES.
- v **TRANSACCIONES ANIDADAS.**
- v SE PRESENTAN CUANDO LAS TRANSACCIONES PUEDEN CONTENER SUBTRANSACCIONES (PROCESOS HIJOS) QUE:
 - v SE EJECUTEN EN PARALELO ENTRE SI EN PROCESADORES DISTINTOS.
 - v PUEDEN ORIGINAR NUEVAS SUBTRANSACCIONES.

IMPLANTACION DEL MODELO DE TRANSACCION

- v EXISTEN VARIOS METODOS DE IMPLANTACION.
- v **ESPACIO DE TRABAJO PARTICULAR.**
- v CONSISTE EN QUE CUANDO UN PROCESO INICIA UNA TRANSACCION SE LE OTORGA UN ESPACIO DE TRABAJO PARTICULAR:
 - u CONTIENE TODOS LOS ARCHIVOS (Y OTROS OBJETOS) A LOS CUALES TIENE ACCESO.
 - u LAS LECTURAS Y ESCRITURAS IRAN A ESTE ESPACIO HASTA QUE LA TRANSACCION SE COMPLETE O ABORTE:
 - Φ EL “ESPACIO REAL” ES EL SISTEMA DE ARCHIVOS NORMAL.
 - u SIGNIFICA ALTO CONSUMO DE RECURSOS POR LAS COPIAS DE LOS OBJETOS AL ESPACIO DE TRABAJO PARTICULAR.
- v CUANDO UN PROCESO INICIA UNA TRANSACCION:
 - u BASTA CREAR UN ESPACIO DE TRABAJO PARTICULAR PARA EL QUE SEA VACIO EXCEPTO POR UN APUNTADOR DE REGRESO AL ESPACIO DE TRABAJO DE SU PROCESO PADRE.
- v PARA UNA TRANSACCION DEL NIVEL SUPERIOR EL ESPACIO DE TRABAJO DEL PADRE ES EL SISTEMA DE ARCHIVOS “REAL”.
- v CUANDO EL PROCESO ABRE UN ARCHIVO PARA LECTURA:
 - u SE SIGUEN LOS APUNTADORES DE REGRESO HASTA LOCALIZAR EL ARCHIVO EN EL ESPACIO DE TRABAJO DEL PADRE (O ALGUN ANTECESOR).
- v CUANDO SE ABRE UN ARCHIVO PARA ESCRITURA:
 - u SE LO LOCALIZA DE MANERA SIMILAR QUE PARA LECTURA.
 - u SE COPIA EN PRIMER LUGAR AL ESPACIO DE TRABAJO PARTICULAR:
 - Φ UNA OPTIMIZACION CONSISTE EN COPIAR SOLO EL INDICE DEL ARCHIVO EN EL ESPACIO DE TRABAJO PARTICULAR:
 - EL INDICE ES EL BLOQUE DE DATOS ASOCIADO A CADA ARCHIVO E INDICA LA LOCALIZACION DE SUS BLOQUES EN EL DISCO.
 - ES EL NODO-I CORRESPONDIENTE.

IMPLANTACION DEL MODELO DE TRANSACCION

- v LA LECTURA POR MEDIO DEL INDICE PARTICULAR (DEL ESPACIO DE TRABAJO PARTICULAR) NO ES PROBLEMÁTICA:
 - υ LAS DIRECCIONES EN DISCO A LAS QUE REFERENCIA SON LAS ORIGINALES.
- v LA MODIFICACION DE UN BLOQUE DE UN ARCHIVO REQUIERE:
 - υ HACER UNA COPIA DEL BLOQUE.
 - υ INSERTAR EN EL INDICE LA DIRECCION DE LA COPIA.
- v LA MODIFICACION SOBRE LA COPIA NO AFECTA AL BLOQUE ORIGINAL.
- v UN TRATAMIENTO SIMILAR SE DA AL AGREGADO DE BLOQUES:
 - υ LOS NUEVOS BLOQUES SE LLAMAN **BLOQUES SOMBRA (SHADOW BLOCKS)**.
- v EL PROCESO QUE EJECUTA LA TRANSACCION VE EL ARCHIVO MODIFICADO PERO LOS DEMAS PROCESOS VEN EL ARCHIVO ORIGINAL.
- v SI LA TRANSACCION ABORTA (TERMINA ANORMALMENTE):
 - υ EL ESPACIO DE TRABAJO PARTICULAR SE ELIMINA.
 - υ LOS BLOQUES PARTICULARES A LOS QUE APUNTA SE COLOCAN NUEVAMENTE EN LA LISTA DE BLOQUES LIBRES.
- v SI LA TRANSACCION SE COMPROMETE (TERMINA NORMALMENTE):
 - υ LOS INDICES PARTICULARES SE DESPLAZAN AL ESPACIO DE TRABAJO DEL PADRE DE MANERA ATOMICA.
 - υ LOS BLOQUES QUE NO SON ALCANZABLES SE COLOCAN EN LA LISTA DE BLOQUES LIBRES.
- v **BITACORA DE ESCRITURA ANTICIPADA.**
- v ESTE METODO TAMBIEN SE DENOMINA **LISTA DE INTENCIONES.**
- v LOS ARCHIVOS REALMENTE SE MODIFICAN PERO ANTES DE CAMBIAR CUALQUIER BLOQUE:
 - υ SE GRABA UN REGISTRO EN LA BITACORA ("LOG") DE ESCRITURA ANTICIPADA EN UN ESPACIO DE ALMACENAMIENTO ESTABLE:
 - ⊕ SE INDICA LA TRANSACCION QUE PRODUCE EL CAMBIO, EL ARCHIVO Y BLOQUE MODIFICADOS Y LOS VALORES ANTERIOR Y NUEVO.

IMPLANTACION DEL MODELO DE TRANSACCION

- v SI LA TRANSACCION TIENE EXITO Y SE HACE UN COMPROMISO:
 - u SE ESCRIBE UN REGISTRO DEL COMPROMISO EN LA BITACORA.
 - u LAS ESTRUCTURAS DE DATOS NO TIENEN QUE MODIFICARSE:
 - Φ YA HAN SIDO ACTUALIZADAS.
- v SI LA TRANSACCION ABORTA:
 - u SE PUEDE UTILIZAR LA BITACORA PARA RESPALDO DEL ESTADO ORIGINAL:
 - Φ A PARTIR DEL FINAL Y HACIA ATRAS:
 - SE LEE CADA REGISTRO DE LA BITACORA.
 - SE DESHACE CADA CAMBIO DESCRIPTO EN EL.
 - u ESTA ACCION SE DENOMINA **RETROALIMENTACION**.
- v POR MEDIO DE LA BITACORA SE PUEDE:
 - u IR HACIA ADELANTE (REALIZAR LA TRANSACCION).
 - u IR HACIA ATRAS (DESHACER LA TRANSACCION).
- v **PROTOCOLO DE COMPROMISO DE DOS FASES (TWO - PHASE COMMIT).**
- v UNO DE LOS PROCESOS QUE INTERVIENEN FUNCIONA COMO EL COORDINADOR.
- v EL COORDINADOR ESCRIBE UNA ENTRADA EN LA BITACORA PARA INDICAR QUE INICIA EL PROTOCOLO.
- v EL COORDINADOR ENVIA A C / U DE LOS PROCESOS RELACIONADOS (SUBORDINADOS) UN MENSAJE PARA QUE ESTEN PREPARADOS PARA EL COMPROMISO.
- v CUANDO UN SUBORDINADO RECIBE EL MENSAJE:
 - u VERIFICA SI ESTA LISTO PARA COMPROMETERSE.
 - u ESCRIBE UNA ENTRADA EN LA BITACORA.
 - u ENVIA DE REGRESO SU DECISION.
- v CUANDO EL COORDINADOR HA RECIBIDO TODAS LAS RESPUESTAS SABE SI DEBE ESTABLECER EL COMPROMISO O ABORTAR:
 - u SI TODOS LOS PROCESOS ESTAN LISTOS PARA COMPROMETERSE CIERRA LA TRANSACCION.
 - u SI ALGUNO DE LOS PROCESOS NO SE COMPROMETE (O NO RESPONDE) LA TRANSACCION SE ABORTA.
 - u EL COORDINADOR:
 - Φ ESCRIBE UNA ENTRADA EN LA BITACORA.
 - Φ ENVIA UN MENSAJE A CADA SUBORDINADO PARA INFORMARLE DE LA DECISION.

CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION

- v LOS ALGORITMOS DE CONTROL DE CONCURRENCIA SON NECESARIOS CUANDO SE EJECUTAN VARIAS TRANSACCIONES DE MANERA SIMULTANEA:
 - u EN DISTINTOS PROCESOS.
 - u EN DISTINTOS PROCESADORES.
- v LOS PRINCIPALES ALGORITMOS SON:
 - u EL DE LA CERRADURA.
 - u EL DEL CONTROL OPTIMISTA DE LA CONCURRENCIA.
 - u EL DE LAS MARCAS DE TIEMPO.
- v **CERRADURA (LOCKING).**
- v CUANDO UN PROCESO DEBE LEER O ESCRIBIR EN UN ARCHIVO (U OTRO OBJETO) COMO PARTE DE UNA TRANSACCION:
 - u PRIMERO CIERRA EL ARCHIVO.
- v LA CERRADURA SE PUEDE HACER MEDIANTE:
 - u UN UNICO MANEJADOR CENTRALIZADO DE CERRADURAS.
 - u UN MANEJADOR LOCAL DE CERRADURAS EN CADA MAQUINA.
- v EL MANEJADOR DE CERRADURAS:
 - u MANTIENE UNA LISTA DE LOS ARCHIVOS CERRADOS.
 - u RECHAZA TODOS LOS INTENTOS POR CERRAR ARCHIVOS YA CERRADOS POR OTROS PROCESOS.
- v EL SISTEMA DE TRANSACCIONES GENERALMENTE ADQUIERE Y LIBERA LAS CERRADURAS SIN ACCION POR PARTE DEL PROGRAMADOR.
- v UNA MEJORA CONSISTE EN DISTINGUIR LAS CERRADURAS PARA LECTURA DE LAS CERRADURAS PARA ESCRITURA.
- v UNA CERRADURA PARA LECTURA NO IMPIDE OTRAS CERRADURAS PARA LECTURA:
 - u LAS CERRADURAS PARA LECTURA SE COMPARTEN.
- v UNA CERRADURA PARA ESCRITURA SI IMPIDE OTRAS CERRADURAS (DE LECTURA O DE ESCRITURA):
 - u LAS CERRADURAS PARA ESCRITURA NO SE COMPARTEN:
 - Φ DEBEN SER EXCLUSIVAS.

CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION

- v EL ELEMENTO POR CERRAR PUEDE SER UN ARCHIVO, UN REGISTRO, UN CAMPO, ETC.:
 - u LO RELATIVO AL TAMAÑO DEL ELEMENTO POR CERRAR SE LLAMA LA **GRANULARIDAD DE LA CERRADURA**.
- v MIENTRAS MAS FINA SEA LA GRANULARIDAD:
 - u PUEDE SER MAS PRECISA LA CERRADURA.
 - u SE PUEDE LOGRAR UN MAYOR PARALELISMO EN EL ACCESO AL RECURSO.
 - u SE REQUIERE UN MAYOR N° DE CERRADURAS.
- v GENERALMENTE SE UTILIZA LA **CERRADURA DE DOS FASES**:
 - u EL PROCESO ADQUIERE TODAS LAS CERRADURAS NECESARIAS DURANTE LA **FASE DE CRECIMIENTO**.
 - u EL PROCESO LAS LIBERA EN LA **FASE DE REDUCCION**.
- v SE DEBEN EVITAR SITUACIONES DE **ABORTO EN CASCADA**:
 - u SE GRABA EN UN ARCHIVO Y LUEGO SE LIBERA SU CERRADURA.
 - u OTRA TRANSACCION LO CIERRA, REALIZA SU TRABAJO Y LUEGO ESTABLECE UN COMPROMISO.
 - u LA TRANSACCION ORIGINAL ABORTA.
 - u LA SEGUNDA TRANSACCION (YA COMPROMETIDA) DEBE DESHACERSE:
 - Φ SUS RESULTADOS SE BASAN EN UN ARCHIVO QUE NO DEBERIA HABER VISTO CUANDO LO HIZO.
- v LAS CERRADURAS COMUNES Y DE DOS FASES PUEDEN PROVOCAR BLOQUEOS CUANDO DOS PROCESOS INTENTAN ADQUIRIR LA MISMA PAREJA DE CERRADURAS PERO EN ORDEN OPUESTO:
 - u SE DEBEN UTILIZAR TECNICAS DE PREVENCION Y DE DETECCION DE BLOQUEOS PARA SUPERAR EL PROBLEMA.

CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION

v CONTROL OPTIMISTA DE LA CONCURRENCIA.

v LA IDEA ES MUY SENCILLA:

- v SE SIGUE ADELANTE Y SE HACE TODO LO QUE SE DEBA HACER, SIN PRESTAR ATENCION A LO QUE HACEN LOS DEMAS.**
- v SE ACTUA A POSTERIORI SI SE PRESENTA ALGUN PROBLEMA.**

v SE MANTIENE UN REGISTRO DE LOS ARCHIVOS LEIDOS O GRABADOS.

v EN EL MOMENTO DEL COMPROMISO:

- v SE VERIFICAN TODAS LAS DEMAS TRANSACCIONES PARA VER SI ALGUNO DE LOS ARCHIVOS HA SIDO MODIFICADO DESDE EL INICIO DE LA TRANSACCION:**
 - Φ SI ESTO OCURRE LA TRANSACCION ABORTA.**
 - Φ SI ESTO NO OCURRE SE REALIZA EL COMPROMISO.**

v LAS PRINCIPALES VENTAJAS SON:

- v AUSENCIA DE BLOQUEOS.**
- v PARALELISMO MAXIMO YA QUE NO SE ESPERAN CERRADURAS.**

v LA PRINCIPAL DESVENTAJA ES:

- v RE-EJECUCION DE LA TRANSACCION EN CASO DE FALLA.**
- v LA PROBABILIDAD DE FALLO PUEDE CRECER SI LA CARGA DE TRABAJO ES MUY ALTA.**

v MARCAS DE TIEMPO.

v SE ASOCIA A CADA TRANSACCION UNA MARCA DE TIEMPO AL INICIAR (*BEGIN_TRANSACTION*).

v SE GARANTIZA QUE LAS MARCAS SON UNICAS MEDIANTE EL ALGORITMO DE LAMPORT.

v CADA ARCHIVO DEL SISTEMA TIENE ASOCIADAS UNA MARCA DE TIEMPO PARA LA LECTURA Y OTRA PARA LA ESCRITURA:

- v INDICAN LA ULTIMA TRANSACCION COMPROMETIDA QUE REALIZO LA LECTURA O ESCRITURA.**

v CUANDO UN PROCESO INTENTE ACCEDER A UN ARCHIVO:

- v LO LOGRARA SI LAS MARCAS DE TIEMPO DE LECTURA Y ESCRITURA SON MENORES (MAS ANTIGUAS) QUE LA MARCA DE LA TRANSACCION ACTIVA.**

CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION

- v SI LA MARCA DE TIEMPO DE LA TRANSACCION ACTIVA ES MENOR QUE LA DEL ARCHIVO QUE INTENTA ACCEDER:
 - u UNA TRANSACCION INICIADA POSTERIORMENTE HA ACCEDIDO AL ARCHIVO Y HA EFECTUADO UN COMPROMISO.
 - u LA TRANSACCION ACTIVA SE HA REALIZADO TARDE Y SE ABORTA.
- v EN EL METODO DE LAS MARCAS NO PREOCUPA QUE LAS TRANSACCIONES CONCURRENTES UTILICEN LOS MISMOS ARCHIVOS:
 - u SI IMPORTA QUE LA TRANSACCION CON EL N° MENOR ESTE EN PRIMER LUGAR.
- v LAS MARCAS DE TIEMPO TIENEN PROPIEDADES DISTINTAS A LAS DE LOS BLOQUEOS:
 - u UNA TRANSACCION ABORTA CUANDO ENCUENTRA UNA MARCA MAYOR (POSTERIOR).
 - u EN IGUALES CIRCUNSTANCIAS Y EN UN ESQUEMA DE CERRADURAS PODRIA ESPERAR O CONTINUAR INMEDIATAMENTE.
- v LAS MARCAS DE TIEMPO SON LIBRES DE BLOQUEOS:
 - u ES UNA GRAN VENTAJA.
- v **RESUMEN.**
- v LOS DIFERENTES ESQUEMAS OFRECEN DISTINTAS VENTAJAS PERO EL PROBLEMA PRINCIPAL ES LA GRAN COMPLEJIDAD DE SU IMPLANTACION.

BLOQUEOS EN SISTEMAS DISTRIBUIDOS

- v SON PEORES QUE LOS BLOQUEOS EN SISTEMAS MONOPROCESADOR:
 - u SON MAS DIFICILES DE EVITAR, PREVENIR, DETECTAR Y SOLUCIONAR.
 - u TODA LA INFORMACION RELEVANTE ESTA DISPERSA EN MUCHAS MAQUINAS.
- v SON ESPECIALMENTE CRITICOS EN SISTEMAS DE BASES DE DATOS DISTRIBUIDOS.
- v LAS **ESTRATEGIAS USUALES** PARA EL MANEJO DE LOS BLOQUEOS SON:
 - u **ALGORITMO DEL AVESTRUZ:**
 - Φ IGNORAR EL PROBLEMAS.
 - u **DETECCION:**
 - Φ PERMITIR QUE OCURRAN LOS BLOQUEOS, DETECTARLOS E INTENTAR RECUPERARSE DE ELLOS.
 - u **PREVENCION:**
 - Φ HACER QUE LOS BLOQUEOS SEAN IMPOSIBLES DESDE EL PUNTO DE VISTA ESTRUCTURAL.
 - u **EVITARLOS:**
 - Φ EVITAR LOS BLOQUEOS MEDIANTE LA ASIGNACION CUIDADOSA DE LOS RECURSOS.
- v EL ALGORITMO DEL AVESTRUZ MERECE LAS MISMAS CONSIDERACIONES QUE EN EL CASO DE MONOPROCESADOR.
- v EN LOS SISTEMAS DISTRIBUIDOS RESULTA MUY DIFICIL IMPLANTAR ALGORITMOS PARA EVITAR LOS BLOQUEOS:
 - u SE REQUIERE SABER DE ANTEMANO LA PROPORCION DE CADA RECURSO QUE NECESITARA CADA PROCESO:
 - Φ ES MUY DIFICIL DISPONER DE ESTA INFORMACION EN FORMA PRACTICA.
- v LAS TECNICAS MAS APLICABLES PARA EL ANALISIS DE LOS BLOQUEOS EN SISTEMAS DISTRIBUIDOS SON:
 - u DETECCION.
 - u PREVENCION.

DETECCION DISTRIBUIDA DE BLOQUEOS

- v CUANDO SE DETECTA UN BLOQUEO EN UN S. O. CONVENCIONAL SE RESUELVE ELIMINANDO UNO O MAS PROCESOS.
- v CUANDO SE DETECTA UN BLOQUEO EN UN SISTEMA BASADO EN TRANSACCIONES ATOMICAS SE RESUELVE ABORTANDO UNA O MAS TRANSACCIONES:
 - v EL SISTEMA RESTAURA EL ESTADO QUE TENIA ANTES DE INICIAR LA TRANSACCION.
 - v LA TRANSACCION PUEDE VOLVER A COMENZAR.
- v LAS CONSECUENCIAS DE LA ELIMINACION DE UN PROCESO:
 - v SON MUCHO MENOS SEVERAS SI SE UTILIZAN LAS TRANSACCIONES QUE EN CASO DE QUE NO SE UTILICEN.
- v **DETECCION CENTRALIZADA DE BLOQUEOS.**
- v CADA MAQUINA MANTIENE LA GRAFICA DE RECURSOS DE SUS PROPIOS PROCESOS Y RECURSOS.
- v UN COORDINADOR CENTRAL MANTIENE LA GRAFICA DE RECURSOS DE TODO EL SISTEMA:
 - v ES LA UNION DE TODAS LAS GRAFICAS INDIVIDUALES.
- v CUANDO EL COORDINADOR DETECTA UN CICLO ELIMINA UNO DE LOS PROCESOS PARA ROMPER EL BLOQUEO.
- v LA INFORMACION DE CONTROL SE DEBE TRANSMITIR EXPLICITAMENTE, EXISTIENDO LAS SIGUIENTES VARIANTES:
 - v CADA MAQUINA INFORMA CADA ACTUALIZACION AL COORDINADOR.
 - v CADA MAQUINA INFORMA PERIODICAMENTE LAS MODIFICACIONES DESDE LA ULTIMA ACTUALIZACION.
 - v EL COORDINADOR REQUIERE LA INFORMACION CUANDO LA NECESITA.
- v LA INFORMACION DE CONTROL INCOMPLETA O RETRASADA PUEDE LLEVAR A **FALSOS BLOQUEOS**:
 - v EL COORDINADOR INTERPRETA ERRONEAMENTE QUE EXISTE UN BLOQUEO Y ELIMINA UN PROCESO.
 - v UNA POSIBLE SOLUCION ES UTILIZAR EL ALGORITMO DE LAMPORT PARA DISPONER DE UN TIEMPO GLOBAL.

DETECCION DISTRIBUIDA DE BLOQUEOS

- v **DETECCION DISTRIBUIDA DE BLOQUEOS.**
- v UN ALGORITMO TIPICO ES EL DE CHANDY-MISRA-HAAS.
- v LOS PROCESOS PUEDEN SOLICITAR VARIOS RECURSOS (POR EJEMPLO CERRADURAS) AL MISMO TIEMPO, EN VEZ DE UNO CADA VEZ.
- v SE PERMITEN LAS SOLICITUDES SIMULTANEAS DE VARIOS PROCESOS:
 - u UN PROCESO PUEDE ESPERAR A UNO O MAS RECURSOS SIMULTANEAMENTE.
 - u LOS RECURSOS QUE ESPERA UN PROCESO PUEDEN SER LOCALES O REMOTOS (DE OTRA MAQUINA).
- v SI EL PROCESO “0” SE BLOQUEA DEBIDO AL PROCESO “1”:
 - u SE GENERA UN MENSAJE DE **EXPLORACION** QUE SE ENVIA AL PROCESO (O PROCESOS) QUE DETIENEN LOS RECURSOS NECESARIOS.
 - u EL MENSAJE CONSTA DE TRES NUMEROS:
 - EL PROCESO RECIEN BLOQUEADO, EL PROCESO QUE ENVIA EL MENSAJE Y EL PROCESO AL CUAL SE ENVIA.
 - u AL LLEGAR EL MENSAJE EL RECEPTOR VERIFICA SI EL MISMO ESPERA A ALGUNOS PROCESOS, EN CUYO CASO:
 - EL MENSAJE SE ACTUALIZA:
 - SE CONSERVA EL PRIMER CAMPO.
 - SE REEMPLAZA EL SEGUNDO POR SU PROPIO N° DE PROCESO Y EL TERCERO POR EL N° DEL PROCESO AL CUAL ESPERA.
 - EL MENSAJE SE ENVIA AL PROCESO DEBIDO AL CUAL SE BLOQUEA:
 - SI SE BLOQUEA DEBIDO A VARIOS PROCESOS LES ENVIA MENSAJES (DIFERENTES) A TODOS ELLOS.
 - u SI UN MENSAJE RECORRE TODO EL CAMINO Y REGRESA A SU EMISOR ORIGINAL (EL PROCESO ENLISTADO EN EL PRIMER CAMPO), ENTONCES:
 - EXISTE UN CICLO Y EL SISTEMA ESTA BLOQUEADO.
- v UNA FORMA DE ROMPER EL BLOQUEO ES QUE EL PROCESO QUE INICIO LA EXPLORACION SE COMPROMETA A SUICIDARSE:
 - u SI VARIOS PROCESOS SE BLOQUEAN AL MISMO TIEMPO E INICIAN EXPLORACIONES, TODOS ELLOS SE SUICIDARAN.
- v UNA VARIANTE ES ELIMINAR SOLO AL PROCESO DEL CICLO QUE TIENE EL N° MAS ALTO.

PREVENCION DISTRIBUIDA DE BLOQUEOS

- v LA PREVENCION CONSISTE EN EL DISEÑO CUIDADOSO DEL SISTEMA PARA QUE LOS BLOQUEOS SEAN IMPOSIBLES ESTRUCTURALMENTE.
- v ENTRE LAS DISTINTAS TECNICAS SE INCLUYE:
 - u PERMITIR A LOS PROCESOS QUE SOLO CONSERVEN UN RECURSO A LA VEZ.
 - u EXIGIR A LOS PROCESOS QUE SOLICITEN TODOS SUS RECURSOS DESDE UN PRINCIPIO.
 - u HACER QUE TODOS LOS PROCESOS LIBEREN TODOS SUS RECURSOS CUANDO SOLICITEN UNO NUEVO.
- v EN UN SISTEMA DISTRIBUIDO CON TIEMPO GLOBAL Y TRANSACCIONES ATOMICAS:
 - u SE PUEDE ASOCIAR A CADA TRANSACCION UNA MARCA DE TIEMPO GLOBAL AL MOMENTO DE SU INICIO.
 - u NO PUEDEN HABER PAREJAS DE TRANSACCIONES CON IGUAL MARCA DE TIEMPO ASOCIADA.
- v LA IDEA ES QUE CUANDO UN PROCESO ESTA A PUNTO DE BLOQUEARSE EN ESPERA DE UN RECURSO QUE ESTA UTILIZANDO OTRO PROCESO:
 - u SE VERIFICA CUAL DE ELLOS TIENE LA MARCA DE TIEMPO MAYOR (ES MAS JOVEN).
 - u SE PUEDE PERMITIR LA ESPERA SOLO SI EL PROCESO EN ESTADO DE ESPERA TIENE UNA MARCA INFERIOR (MAS VIEJO) QUE EL OTRO.
- v AL SEGUIR CUALQUIER CADENA DE PROCESOS EN ESPERA:
 - u LAS MARCAS APARECEN EN FORMA CRECIENTE.
 - u LOS CICLOS SON IMPOSIBLES.
- v OTRA POSIBILIDAD ES PERMITIR LA ESPERA DE PROCESOS SOLO SI EL PROCESO QUE ESPERA TIENE UNA MARCA MAYOR (ES MAS JOVEN) QUE EL OTRO PROCESO:
 - u LAS MARCAS APARECEN EN LA CADENA EN FORMA DESCENDENTE.
- v ES MAS SABIO DAR PRIORIDAD A LOS PROCESOS MAS VIEJOS:
 - u SE HA INVERTIDO TIEMPO DE PROCESO EN ELLOS.
 - u PROBABLEMENTE CONSERVAN MAS RECURSOS.

PROCESOS Y PROCESADORES EN SISTEMAS DISTRIBUIDOS

- v INTRODUCCION A LOS HILOS (THREADS)
- v USO DE HILOS
- v ASPECTOS DEL DISEÑO DE UN PAQUETE DE HILOS
- v IMPLANTACION DE UN PAQUETE DE HILOS
- v HILOS Y RPC
- v MODELOS DE SISTEMAS
- v EL MODELO DE ESTACION DE TRABAJO
- v USO DE ESTACIONES DE TRABAJO INACTIVAS
- v EL MODELO DE LA PILA DE PROCESADORES
- v ASIGNACION DE PROCESADORES
- v MODELOS DE ASIGNACION
- v ASPECTOS DEL DISEÑO DE ALGORITMOS DE ASIGNACION DE PROCESADORES
- v ASPECTOS DE LA IMPLANTACION DE ALGORITMOS DE ASIGNACION DE PROCESADORES
- v EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES
- v PLANIFICACION EN SISTEMAS DISTRIBUIDOS

INTRODUCCION A LOS HILOS (THREADS)

- v MUCHOS S. O. DISTRIBUIDOS SOPORTAN MULTIPLES HILOS DE CONTROL DENTRO DE UN PROCESO QUE:
 - υ COMPARTEN UN UNICO ESPACIO DE DIRECCIONES.
 - υ SE EJECUTAN QUASI - PARALELAMENTE COMO SI FUERAN PROCESOS INDEPENDIENTES.
- v EJ.: SERVIDOR DE ARCHIVOS QUE DEBE BLOQUEARSE OCASIONALMENTE EN ESPERA DE ACCESO AL DISCO:
 - υ SI TIENE VARIOS HILOS DE CONTROL PODRIA EJECUTAR UN SEGUNDO HILO MIENTRAS EL PRIMERO ESPERA:
 - Φ EL RESULTADO SERIA MEJOR RENDIMIENTO Y DESEMPEÑO.
 - Φ NO SE LOGRA ESTO CON PROCESOS SERVIDORES INDEPENDIENTES PUESTO QUE DEBEN COMPARTIR UN BUFFER CACHE COMUN:
 - DEBEN ESTAR EN EL MISMO ESPACIO DE DIRECCIONES.
- v EN MUCHOS SENTIDOS LOS HILOS SON COMO MINIPROCESOS:
 - υ CADA HILO:
 - Φ SE EJECUTA EN FORMA EstrictAMENTE SECUENCIAL.
 - Φ TIENE SU PROPIO CONTADOR DE PROGRAMA Y UNA PILA PARA LLEVAR UN REGISTRO DE SU POSICION.
 - υ LOS HILOS COMPARTEN LA CPU DE LA MISMA FORMA QUE LO HACEN LOS PROCESOS:
 - Φ SECUENCIALMENTE, EN TIEMPO COMPARTIDO.
 - υ SOLO EN UN MULTIPROCESADOR SE PUEDEN EJECUTAR REALMENTE EN PARALELO.
 - υ LOS HILOS PUEDEN CREAR HILOS HIJOS.
 - υ MIENTRAS UN HILO ESTA BLOQUEADO SE PUEDE EJECUTAR OTRO HILO DEL MISMO PROCESO.
- v LOS DISTINTOS HILOS DE UN PROCESO COMPARTEN:
 - υ UN ESPACIO DE DIRECCIONES, EL CONJUNTO DE ARCHIVOS ABIERTOS, LOS PROCESOS HIJOS, CRONOMETROS, SEÑALES, ETC.
- v LOS HILOS PUEDEN TENER DISTINTOS ESTADOS:
 - υ EN EJECUCION, BLOQUEADO, LISTO, TERMINADO.

USO DE HILOS

- v LOS HILOS PERMITEN LA COMBINACION DEL PARALELISMO CON LA EJECUCION SECUENCIAL Y EL BLOQUEO DE LAS LLAMADAS AL SISTEMA.
- v CONSIDERAMOS EL *EJEMPLO DEL SERVIDOR DE ARCHIVOS* CON SUS POSIBLES ORGANIZACIONES PARA MUCHOS HILOS DE EJECUCION.
- v INICIAMOS CON EL **MODELO SERVIDOR / TRABAJADOR**:
 - u UN HILO, EL **SERVIDOR**, LEE LAS SOLICITUDES DE TRABAJO EN EL BUZON DEL SISTEMA.
 - u ELIGE A UN **HILO TRABAJADOR** INACTIVO (BLOQUEADO) Y LE ENVIA LA SOLICITUD, DESPERTANDOLO.
 - u EL H. TRABAJADOR VERIFICA SI PUEDE SATISFACER LA SOLICITUD POR MEDIO DEL BLOQUE CACHE COMPARTIDO, AL QUE TIENEN ACCESO TODOS LOS HILOS.
 - u SI NO ENVIA UN MENSAJE AL DISCO PARA OBTENER EL BLOQUE NECESARIO Y SE DUERME ESPERANDO EL FIN DE LA OPERACION.
 - u SE LLAMA:
 - Φ AL PLANIFICADOR Y SE INICIALIZA OTRO HILO, QUE TAL VEZ SEA EL SERVIDOR, PARA PEDIR MAS TRABAJO; O.
 - Φ A OTRO TRABAJADOR LISTO PARA REALIZAR UN TRABAJO.
- v LOS HILOS GANAN UN DESEMPEÑO CONSIDERABLE PERO C / U DE ELLOS SE PROGRAMA EN FORMA SECUENCIAL.
- v OTRO MODELO ES EL DE **EQUIPO**:
 - u TODOS LOS HILOS SON IGUALES Y C / U OBTIENE Y PROCESA SUS PROPIAS SOLICITUDES.
 - u NO HAY SERVIDOR.
 - u SE UTILIZA UNA COLA DE TRABAJO QUE CONTIENE TODOS LOS TRABAJOS PENDIENTES:
 - Φ SON TRABAJOS QUE LOS HILOS NO HAN PODIDO MANEJAR.
 - u UN HILO DEBE VERIFICAR PRIMERO LA COLA DE TRABAJO ANTES DE BUSCAR EN EL BUZON DEL SISTEMA.
- v UN TERCER MODELO ES EL DE **ENTUBAMIENTO**:
 - u EL PRIMER HILO GENERA CIERTOS DATOS Y LOS TRANSFIERE AL SIGUIENTE PARA SU PROCESAMIENTO.
 - u LOS DATOS PASAN DE HILO EN HILO Y EN CADA ETAPA SE LLEVA A CABO CIERTO PROCESAMIENTO.
- v UN PROGRAMA DISEÑADO ADECUADAMENTE Y QUE UTILICE HILOS DEBE FUNCIONAR BIEN:
 - u EN UNA UNICA CPU CON HILOS COMPARTIDOS.
 - u EN UN VERDADERO MULTIPROCESADOR.

ASPECTOS DEL DISEÑO DE UN PAQUETE DE HILOS

- v UN CONJUNTO DE PRIMITIVAS RELACIONADAS CON LOS HILOS (EJ.: LLAMADAS A BIBLIOTECA) DISPONIBLES PARA LOS USUARIOS SE LLAMA UN **PAQUETE DE HILOS**.
- v RESPECTO DEL *MANEJO DE LOS HILOS* SE TIENEN HILOS **ESTATICOS** E HILOS **DINAMICOS**.
- v EN UN **DISEÑO ESTATICO**:
 - u SE ELIGE EL N° DE HILOS AL ESCRIBIR EL PROGRAMA O DURANTE SU COMPILACION.
 - u C / U DE ELLOS TIENE ASOCIADA UNA PILA FIJA.
 - u SE LOGRA SIMPLICIDAD PERO TAMBIEN INFLEXIBILIDAD.
- v EN UN **DISEÑO DINAMICO**:
 - u SE PERMITE LA CREACION Y DESTRUCCION DE LOS HILOS DURANTE LA EJECUCION.
 - u LA LLAMADA PARA LA CREACION DE HILOS DETERMINA:
 - Φ EL PROGRAMA PRINCIPAL DEL HILO.
 - Φ UN TAMAÑO DE PILA.
 - Φ UNA PRIORIDAD DE PLANIFICACION, ETC.
 - u LA LLAMADA GENERALMENTE REGRESA UN IDENTIFICADOR DE HILO:
 - Φ SE USARA EN LAS POSTERIORES LLAMADAS RELACIONADAS AL HILO.
 - u UN PROCESO:
 - Φ SE INICIA CON UN SOLO HILO.
 - Φ PUEDE CREAR EL N° NECESARIO DE HILOS.
- v LOS HILOS PUEDEN CONCLUIR:
 - u POR SU CUENTA, AL TERMINAR SU TRABAJO.
 - u POR SU ELIMINACION DESDE EL EXTERIOR.
- v LOS HILOS COMPARTEN UNA MEMORIA COMUN:
 - u CONTIENE DATOS QUE LOS DISTINTOS HILOS COMPARTEN.
 - u EL ACCESO GENERALMENTE SE CONTROLA MEDIANTE *REGIONES CRITICAS*.

IMPLANTACION DE UN PAQUETE DE HILOS

- v UN PAQUETE DE HILOS SE PUEDE IMPLANTAR EN EL ESPACIO:
 - u DEL USUARIO.
 - u DEL NUCLEO.
- v IMPLANTACION DEL PAQUETE DE HILOS EN EL *ESPACIO DEL USUARIO*:
 - u EL NUCLEO NO SABE DE SU EXISTENCIA.
 - u EL NUCLEO MANEJA PROCESOS CON UN UNICO HILO.
 - u NO REQUIERE SOPORTE DE HILOS POR PARTE DEL S. O.
 - u LOS HILOS SE EJECUTAN EN UN SISTEMA DE TIEMPO DE EJECUCION:
 - ⊕ ES UN GRUPO DE PROCEDIMIENTOS QUE MANEJAN LOS HILOS.
 - u CUANDO UN HILO EJECUTA UNA LLAMADA AL SISTEMA O CUALQUIER ACCION QUE PUEDA PROVOCAR SU SUSPENSION:
 - ⊕ LLAMA A UN PROCEDIMIENTO DEL SISTEMA DE TIEMPO DE EJECUCION.
 - ⊕ EL PROCEDIMIENTO VERIFICA SI HAY QUE SUSPENDER AL HILO, EN CUYO CASO:
 - ALMACENA LOS REGISTROS DEL HILO EN UNA TABLA.
 - BUSCA UN HILO NO BLOQUEADO PARA EJECUTARLO.
 - VUELVE A CARGAR LOS REGISTROS DE LA MAQUINA CON LOS VALORES RESGUARDADOS DEL NUEVO HILO.
 - u LAS PRINCIPALES VENTAJAS SON:
 - ⊕ EL INTERCAMBIO DE HILOS ES MAS RAPIDO QUE SI SE UTILIZARAN LOS SEÑALAMIENTOS AL NUCLEO.
 - ⊕ CADA PROCESO PUEDE TENER SU PROPIO ALGORITMO ADAPTADO DE PLANIFICACION DE HILOS.
 - ⊕ TIENEN UNA MEJOR ESCALABILIDAD PARA UN N° MUY GRANDE DE HILOS, YA QUE NO AFECTAN AL NUCLEO CON TABLAS Y BLOQUES DE CONTROL (PILA).

IMPLANTACION DE UN PAQUETE DE HILOS

- v IMPLANTACION DEL PAQUETE DE HILOS EN EL *ESPACIO DEL NUCLEO*:
 - u NO SE NECESITA UN SISTEMA DE TIEMPO DE EJECUCION.
 - u PARA CADA PROCESO EL NUCLEO TIENE UNA TABLA CON UNA ENTRADA POR CADA HILO QUE CONTIENE:
 - Φ LOS REGISTROS, ESTADOS, PRIORIDADES Y DEMAS INFORMACION RELATIVA AL HILO.
 - u TODAS LAS LLAMADAS QUE PUEDEN BLOQUEAR UN HILO SE IMPLANTAN COMO LLAMADAS AL SISTEMA:
 - Φ SIGNIFICA UN COSTO MAYOR (EN RECURSOS Y TIEMPO).
 - u CUANDO UN HILO SE BLOQUEA, EL NUCLEO PUEDE EJECUTAR:
 - Φ OTRO HILO LISTO DEL MISMO PROCESO.
 - Φ UN HILO DE OTRO PROCESO:
 - CON LOS HILOS A NIVEL USUARIO EL SISTEMA DE TIEMPO DE EJECUCION MANTIENE EN EJECUCION LOS HILOS DE SU PROPIO PROCESO HASTA QUE:
 - EL NUCLEO LES RETIRA LA CPU, O.
 - NO HAY HILOS LISTOS.
- v UN PROBLEMA FUNDAMENTAL DE LOS PAQUETES DE HILOS A NIVEL USUARIO ES EL DE LAS LLAMADAS AL SISTEMA CON BLOQUEO:
 - u NO SE PUEDE PERMITIR QUE EL HILO REALMENTE REALICE LA LLAMADA AL SISTEMA:
 - Φ DETENDRIA A TODOS LOS HILOS DEL PROCESO.
 - Φ UN HILO BLOQUEADO NO DEBE AFECTAR A LOS DEMAS.
 - u UNA SOLUCION ES AGREGAR CODIGO JUNTO A LA LLAMADA AL SISTEMA PARA VERIFICAR SI LA MISMA NO GENERARIA BLOQUEO:
 - Φ SE EFECTUARIA LA LLAMADA AL SISTEMA SOLO SI LA VERIFICACION DA O.K.
 - Φ EL CODIGO ADICIONAL SUELE LLAMARSE **JACKET**.
- v OTRO PROBLEMA DE LOS PAQUETES DE HILOS A NIVEL USUARIO ES QUE SI UN HILO COMIENZA SU EJECUCION NO PUEDE EJECUTARSE NINGUN OTRO HILO DE ESE PROCESO:
 - u SALVO QUE EL HILO ENTREGUE VOLUNTARIAMENTE LA CPU.
- v UN PROBLEMA ADICIONAL PARA LOS HILOS A NIVEL USUARIO ES QUE GENERALMENTE LOS PROGRAMADORES DESEAN LOS HILOS EN APLICACIONES DONDE LOS HILOS SE BLOQUEAN A MENUDO:
 - u EJ.: SERVIDOR DE ARCHIVOS CON VARIOS HILOS.

HILOS Y RPC

- v ES COMUN QUE LOS SISTEMAS DISTRIBUIDOS UTILICEN RPC E HILOS.
- v AL INICIAR UN HILO SERVIDOR, S, ESTE EXPORTA SU INTERFAZ AL INFORMARLE DE ESTA AL NUCLEO:
 - u LA INTERFAZ DEFINE LOS PROCEDIMIENTOS QUE PUEDE LLAMAR, SUS PARAMETROS, ETC.
- v AL INICIAR UN HILO CLIENTE, C, ESTE IMPORTA LA INTERFAZ DEL NUCLEO:
 - u SE LE PROPORCIONA UN IDENTIFICADOR ESPECIAL PARA UTILIZARLO EN LA LLAMADA.
 - u EL NUCLEO SABE QUE C LLAMARA POSTERIORMENTE A S:
 - Φ CREA ESTRUCTURAS DE DATOS ESPECIALES PARA PREPARARSE PARA LA LLAMADA.
- v UNA DE LAS ESTRUCTURAS ES UN PILA DE ARGUMENTOS COMPARTIDA POR C Y S:
 - u SE ASOCIA DE MANERA LECTURA / ESCRITURA EN AMBOS ESPACIOS DE DIRECCIONES.
- v PARA LLAMAR AL SERVIDOR, C:
 - u COLOCA SUS ARGUMENTOS EN LA PILA COMPARTIDA MEDIANTE EL PROCEDIMIENTO NORMAL DE TRANSFERENCIA.
 - u HACE UN SEÑALAMIENTO AL NUCLEO COLOCANDO UN IDENTIFICADOR ESPECIAL EN UN REGISTRO.
- v EL NUCLEO:
 - u DETECTA ESTO Y DEDUCE QUE ES UNA LLAMADA LOCAL.
 - u MODIFICA EL MAPA DE MEMORIA DEL CLIENTE PARA COLOCAR ESTE EN EL ESPACIO DE DIRECCIONES DEL SERVIDOR.
 - u INICIA EL HILO CLIENTE, AL EJECUTAR EL PROCEDIMIENTO DEL SERVIDOR.
- v LA LLAMADA SE EFECTUA DE TAL FORMA QUE:
 - u LOS ARGUMENTOS SE ENCUENTRAN YA EN SU LUGAR:
 - Φ NO ES NECESARIO SU COPIADO U ORDENAMIENTO:
 - LA RPC LOCAL SE PUEDE REALIZAR MAS RAPIDO DE ESTA MANERA.

MODELOS DE SISTEMAS

EL MODELO DE ESTACION DE TRABAJO

- v **MODELOS DE SISTEMAS.**
- v EN UN SISTEMA DISTRIBUIDO, CON VARIOS PROCESADORES, UN ASPECTO FUNDAMENTAL DEL DISEÑO ES COMO SE LOS UTILIZA.
- v LOS PROCESADORES DISTRIBUIDOS SE PUEDEN ORGANIZAR DE VARIAS FORMAS:
 - υ MODELO DE ESTACION DE TRABAJO.
 - υ MODELO DE LA PILA DE PROCESADORES.
 - υ MODELO HIBRIDO.
- v **EL MODELO DE ESTACION DE TRABAJO.**
- v EL SISTEMA CONSTA DE ESTACIONES DE TRABAJO (PC) DISPERSAS CONECTADAS ENTRE SI MEDIANTE UNA RED DE AREA LOCAL (LAN).
- v PUEDEN CONTAR O NO CON DISCO RIGIDO EN C / U DE ELLAS.
- v LOS USUARIOS TIENEN:
 - υ UNA CANTIDAD FIJA DE PODER DE COMPUTO EXCLUSIVA.
 - υ UN ALTO GRADO DE AUTONOMIA PARA ASIGNAR LOS RECURSOS DE SU ESTACION DE TRABAJO.

EL MODELO DE ESTACION DE TRABAJO

- v USO DE LOS DISCOS EN LAS ESTACIONES DE TRABAJO:
 - u SIN DISCO:
 - Φ BAJO COSTO, FACIL MANTENIMIENTO DEL HARDWARE Y DEL SOFTWARE, SIMETRIA Y FLEXIBILIDAD.
 - Φ GRAN USO DE LA RED, LOS SERVIDORES DE ARCHIVOS SE PUEDEN CONVERTIR EN CUELLOS DE BOTELLA.
 - u DISCO PARA PAGINACION Y ARCHIVOS DE TIPO BORRADOR:
 - Φ REDUCE LA CARGA DE LA RED RESPECTO DEL CASO ANTERIOR.
 - Φ ALTO COSTO DEBIDO AL GRAN N° DE DISCOS NECESARIOS.
 - u DISCO PARA PAGINACION, ARCHIVOS DE TIPO BORRADOR Y ARCHIVOS BINARIOS (EJECUTABLES):
 - Φ REDUCE AUN MAS LA CARGA SOBRE LA RED.
 - Φ ALTO COSTO Y COMPLEJIDAD ADICIONAL PARA ACTUALIZAR LOS BINARIOS.
 - u DISCO PARA PAGINACION, BORRADOR, BINARIOS Y OCULTAMIENTO DE ARCHIVOS:
 - Φ REDUCE AUN MAS LA CARGA DE RED Y DE LOS SERVIDORES DE ARCHIVOS.
 - Φ ALTO COSTO.
 - Φ PROBLEMAS DE CONSISTENCIA DEL CACHE.
 - u SISTEMA LOCAL DE ARCHIVOS COMPLETO:
 - Φ ESCASA CARGA EN LA RED.
 - Φ ELIMINA LA NECESIDAD DE LOS SERVIDORES DE ARCHIVOS.
 - Φ PERDIDA DE TRANSPARENCIA.

USO DE ESTACIONES DE TRABAJO INACTIVAS

- v LA IDEA CONSISTE EN ORDENAR REMOTAMENTE LA EJECUCION DE PROCESOS EN ESTACIONES DE TRABAJO INACTIVAS.
- v LOS ASPECTOS CLAVE SON:
 - u ¿COMO ENCONTRAR UNA ESTACION DE TRABAJO INACTIVA?.
 - u ¿COMO LOGRAR QUE UN PROCESO REMOTO SE EJECUTE DE FORMA TRANSPARENTE?.
 - u ¿QUE OCURRE SI REGRESA EL POSEEDOR DE LA MAQUINA?.
- v GENERALMENTE SE CONSIDERA QUE UNA ESTACION DE TRABAJO ESTA “INACTIVA” CUANDO SE DAN AMBAS CONDICIONES:
 - u NADIE TOCA EL RATON O EL TECLADO DURANTE VARIOS MINUTOS.
 - u NO SE EJECUTA ALGUN PROCESO INICIADO POR EL USUARIO.
- v LOS ALGORITMOS PARA LOCALIZAR LAS ESTACIONES DE TRABAJO INACTIVAS SE PUEDEN DIVIDIR EN DOS CATEGORIAS:
 - u CONTROLADOS POR EL SERVIDOR.
 - u CONTROLADOS POR EL CLIENTE.
- v *ALGORITMOS CONTROLADOS POR EL SERVIDOR:*
 - u CUANDO UNA ESTACION DE TRABAJO ESTA INACTIVA:
 - Φ SE CONVIERTE EN UN SERVIDOR POTENCIAL.
 - Φ ANUNCIA SU DISPONIBILIDAD:
 - PROPORCIONA SU NOMBRE, DIRECCION EN LA RED Y PROPIEDADES:
 - GRABANDOLOS EN UN ARCHIVO, O.
 - TRANSMITIENDOLOS A LAS OTRAS ESTACIONES.
 - u SE PUEDEN DAR SITUACIONES DE COMPETENCIA ENTRE DISTINTOS USUARIOS PARA ACCEDER A LA MISMA ESTACION INACTIVA AL MISMO TIEMPO:
 - Φ SE DEBEN DETECTAR AL INGRESAR EL REQUERIMIENTO.
 - Φ SOLO PROGRESA EL PRIMER REQUERIMIENTO ARRIBADO.
 - Φ SE ELIMINA A LA ESTACION DE LA LISTA DE INACTIVAS.
 - Φ QUIEN HIZO EL LLAMADO PUEDE ENVIAR SU AMBIENTE E INICIAR EL PROCESO REMOTO.

USO DE ESTACIONES DE TRABAJO INACTIVAS

v *ALGORITMOS CONTROLADOS POR EL CLIENTE:*

- v EL CLIENTE TRANSMITE UNA SOLICITUD INDICANDO EL PROGRAMA QUE DESEA EJECUTAR, LA CANTIDAD DE MEMORIA NECESARIA, SI REQUIERE UN CHIP COPROCESADOR, ETC.
- v AL REGRESAR LA RESPUESTA SE ELIGE UNA ESTACION Y SE LA CONFIGURA.
- v **PARA EJECUTAR EL PROCESO EN LA ESTACION REMOTA SELECCIONADA SE DEBE LOGRAR:**
 - v EL DESPLAZAMIENTO DEL CODIGO.
 - v LA CONFIGURACION DEL PROCESO REMOTO DE MODO QUE:
 - Φ “VEA” EL MISMO AMBIENTE QUE TENDRIA EN EL CASO LOCAL, EN LA **ESTACION DE TRABAJO DE ORIGEN**.
 - Φ EJECUTE DE LA MISMA FORMA QUE EN EL CASO LOCAL.
- v SE NECESITA LA MISMA VISION DEL SISTEMA DE ARCHIVOS, EL MISMO DIRECTORIO DE TRABAJO, ETC.
- v SI SE TRABAJA SOBRE EL SERVIDOR DE ARCHIVOS SE ENVIAN LAS SOLICITUDES DE DISCO AL SERVIDOR.
- v SI SE TRABAJA CON DISCOS LOCALES SE ENVIAN LAS SOLICITUDES A LA MAQUINA DE ORIGEN PARA SU EJECUCION.
- v CIERTAS OPERACIONES COMO LA LECTURA DEL TECLADO Y LA ESCRITURA EN LA PANTALLA:
 - v NUNCA SE PUEDEN EJECUTAR EN LA MAQUINA REMOTA.
 - v DEBEN REGRESAR A LA MAQUINA DE ORIGEN.
- v TODAS LAS LLAMADAS AL SISTEMA QUE SOLICITEN EL ESTADO DE LA MAQUINA DEBEN REALIZARSE EN LA MAQUINA DONDE SE EJECUTA EL PROCESO.
- v LAS LLAMADAS AL SISTEMA RELACIONADAS CON EL TIEMPO SON UN SERIO PROBLEMA DEBIDO A LAS DIFICULTADES DE SINCRONIZACION.
- v EN CASO DE QUE **REGRESE EL POSEEDOR DE LA MAQUINA:**
 - v SE PODRIA NO HACER NADA, CONTRA LA IDEA DE ESTACIONES DE TRABAJO “PERSONALES”.
 - v SE PODRIA ELIMINAR EL PROCESO INTRUSO:
 - Φ ABRUPTAMENTE, PERDIENDOSE EL TRABAJO HECHO Y GENERANDO CAOS EN EL SISTEMA DE ARCHIVOS.
 - Φ ORDENADAMENTE, SALVANDO EL PROCESAMIENTO YA HECHO Y PRESERVANDO LA INTEGRIDAD DEL SISTEMA DE ARCHIVOS.
 - v SE PODRIA EMIGRAR EL PROCESO A OTRA ESTACION.

EL MODELO DE LA PILA DE PROCESADORES

- v SE DISPONE DE UN CONJUNTO DE CPU QUE SE PUEDEN ASIGNAR DINAMICAMENTE A LOS USUARIOS SEGUN LA DEMANDA.
- v LOS USUARIOS NO DISPONEN DE ESTACIONES DE TRABAJO SINO DE TERMINALES GRAFICAS DE ALTO RENDIMIENTO.
- v NO EXISTE EL CONCEPTO DE PROPIEDAD DE LOS PROCESADORES, LOS QUE PERTENECEN A TODOS Y SE UTILIZAN COMPARTIDAMENTE.
- v EL PRINCIPAL ARGUMENTO PARA LA CENTRALIZACION DEL PODER DE COMPUTO COMO UNA PILA DE PROCESADORES PROVIENE DE LA TEORIA DE COLAS:
 - u LLAMAMOS “ λ ” A LA TASA DE ENTRADAS TOTALES DE SOLICITUDES POR SEGUNDO DE TODOS LOS USUARIOS COMBINADOS.
 - u LLAMAMOS “ μ ” A LA TASA DE PROCESAMIENTO DE SOLICITUDES POR PARTE DEL SERVIDOR.
 - u PARA UNA OPERACION ESTABLE DEBE DARSE QUE “ $\mu > \lambda$ ”:
 - Φ SE PUEDEN PERMITIR PEQUEÑOS LAPROS DE TIEMPO EN LOS QUE LA TASA DE ENTRADA EXCEDA A LA DE SERVICIO.
 - u LLAMAMOS “T” AL PROMEDIO DE TIEMPO ENTRE LA EMISION DE UNA SOLICITUD Y LA OBTENCION DE UNA RESPUESTA COMPLETA:
 - Φ $T = 1 / (\mu - \lambda)$.
 - Φ CUANDO “ λ ” TIENDE A “0” “T” NO TIENDE A “0”.
 - u SUPONGAMOS QUE TENEMOS “n” MULTIPROCESADORES PERSONALES, C / U CON CIERTO N° DE CPU Y CON SU PROPIO SISTEMA DE COLAS CON TASAS “ λ ” Y “ μ ” Y TIEMPO “T”:
 - Φ SI REUNIMOS TODAS LAS CPU Y FORMAMOS UNA SOLA PILA DE PROCESADORES TENDREMOS UN SOLO SISTEMA DE COLAS EN VEZ DE “n” COLAS EJECUTANDOSE EN PARALELO.
 - Φ LA TASA DE ENTRADA SERA “ $n \lambda$ ”, LA TASA DE SERVICIO SERA “ $n \mu$ ” Y EL TIEMPO PROMEDIO DE RESPUESTA SERA:
 - $T_1 = 1 / (n \mu - n \lambda) = 1 / n (\mu - \lambda) = T / n$.
 - Φ CONCLUSION: SI REEMPLAZAMOS “n” PEQUEÑOS RECURSOS POR UNO GRANDE QUE SEA “n” VECES MAS PODEROSO:
 - PODEMOS REDUCIR EL TIEMPO PROMEDIO DE RESPUESTA “n” VECES.
- v EL MODELO DE PILA ES MAS EFICIENTE QUE EL MODELO DE BUSQUEDA DE ESTACIONES INACTIVAS.
- v TAMBIEN EXISTE EL **MODELO HIBRIDO** QUE CONSTA DE ESTACIONES DE TRABAJO Y UNA PILA DE PROCESADORES.

ASIGNACION DE PROCESADORES

MODELOS DE ASIGNACION

- v **ASIGNACION DE PROCESADORES.**
- v SON NECESARIOS ALGORITMOS PARA DECIDIR CUAL PROCESO HAY QUE EJECUTAR Y EN QUE MAQUINA.
- v PARA EL MODELO DE ESTACIONES DE TRABAJO:
 - u DECIDIR CUANDO EJECUTAR EL PROCESO DE MANERA LOCAL Y CUANDO BUSCAR UNA ESTACION INACTIVA.
- v PARA EL MODELO DE LA PILA DE PROCESADORES:
 - u DECIDIR DONDE EJECUTAR CADA NUEVO PROCESO.
- v **MODELOS DE ASIGNACION.**
- v GENERALMENTE SE UTILIZAN LAS SIGUIENTES HIPOTESIS:
 - u TODAS LAS MAQUINAS SON IDENTICAS (O AL MENOS COMPATIBLES EN EL CODIGO):
 - Φ DIFIEREN A LO SUMO EN LA VELOCIDAD.
 - u CADA PROCESADOR SE PUEDE COMUNICAR CON LOS DEMAS.
- v LAS ESTRATEGIAS DE ASIGNACION DE PROCESADORES SE DIVIDEN EN:
 - u **NO MIGRATORIAS:**
 - Φ UNA VEZ COLOCADO UN PROCESO EN UNA MAQUINA PERMANECE AHI HASTA QUE TERMINA.
 - u **MIGRATORIAS:**
 - Φ UN PROCESO SE PUEDE TRASLADAR AUNQUE HAYA INICIADO SU EJECUCION.
 - Φ PERMITEN UN MEJOR BALANCE DE LA CARGA PERO SON MAS COMPLEJAS.
- v LOS ALGORITMOS DE ASIGNACION INTENTAN OPTIMIZAR ALGO:
 - u **USO DE LAS CPU:**
 - Φ MAXIMIZAR EL N° DE CICLOS DE CPU QUE SE EJECUTAN PARA TRABAJOS DE LOS USUARIOS.
 - Φ MINIMIZAR EL TIEMPO DE INACTIVIDAD DE LAS CPU.
 - u **TIEMPO PROMEDIO DE RESPUESTA:**
 - Φ MINIMIZAR NO LOS TIEMPOS INDIVIDUALES DE RESPUESTA SINO LOS TIEMPOS PROMEDIO DE RESPUESTA.
 - u **TASA DE RESPUESTA:**
 - Φ MINIMIZAR LA TASA DE RESPUESTA:
 - TIEMPO NECESARIO PARA EJECUTAR UN PROCESO EN CIERTA MAQUINA DIVIDIDO POR EL TIEMPO QUE TARDARIA EN CIERTO PROCESADOR DE REFERENCIA.

ASPECTOS DEL DISEÑO DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v LOS PRINCIPALES ASPECTOS SON LOS SIGUIENTES:
 - u ALGORITMOS DETERMINISTAS VS. HEURISTICOS.
 - u ALGORITMOS CENTRALIZADOS VS. DISTRIBUIDOS.
 - u ALGORITMOS OPTIMOS VS. SUBOPTIMOS.
 - u ALGORITMOS LOCALES VS. GLOBALES.
 - u ALGORITMOS INICIADOS POR EL EMISOR VS. INICIADOS POR EL RECEPTOR.
- v LOS ALGORITMOS DETERMINISTAS SON ADECUADOS CUANDO SE SABE ANTICIPADAMENTE TODO ACERCA DEL COMPORTAMIENTO DE LOS PROCESOS:
 - u ESTO GENERALMENTE NO SE DA, AUNQUE PUEDE HABER EN CIERTOS CASOS APROXIMACIONES ESTADISTICAS.
- v LOS ALGORITMOS HEURISTICOS SON ADECUADOS CUANDO LA CARGA ES IMPREDECIBLE.
- v LOS DISEÑOS CENTRALIZADOS PERMITEN REUNIR TODA LA INFORMACION EN UN LUGAR Y TOMAR UNA MEJOR DECISION:
 - u LA DESVENTAJA ES QUE LA MAQUINA CENTRAL SE PUEDE SOBRECARGAR Y SE PIERDE ROBUSTEZ ANTE SU POSIBLE FALLA.
- v GENERALMENTE LOS ALGORITMOS OPTIMOS CONSUMEN MAS RECURSOS QUE LOS SUBOPTIMOS:
 - u EN LA MAYORIA DE LOS SISTEMAS REALES SE BUSCAN SOLUCIONES SUBOPTIMAS, HEURISTICAS Y DISTRIBUIDAS.
- v CUANDO SE VA A CREAR UN PROCESO SE DEBE DECIDIR SI SE EJECUTARA EN LA MAQUINA QUE LO GENERA O EN OTRA (**POLITICA DE TRANSFERENCIA**):
 - u LA DECISION SE PUEDE TOMAR “SOLO CON INFORMACION LOCAL” O “CON INFORMACION GLOBAL”.
 - u LOS ALGORITMOS LOCALES SON SENCILLOS PERO NO OPTIMOS.
 - u LOS ALGORITMOS GLOBALES SON MEJORES PERO CONSUMEN MUCHOS RECURSOS.
- v CUANDO UNA MAQUINA SE DESHACE DE UN PROCESO LA **POLITICA DE LOCALIZACION** DEBE DECIDIR DONDE ENVIARLO:
 - u NECESITA INFORMACION DE LA CARGA EN TODAS PARTES, OBTENIENDOLA DE:
 - Φ UN EMISOR SOBRECARGADO QUE BUSCA UNA MAQUINA INACTIVA.
 - Φ UN RECEPTOR DESOCUPADO QUE BUSCA TRABAJO.

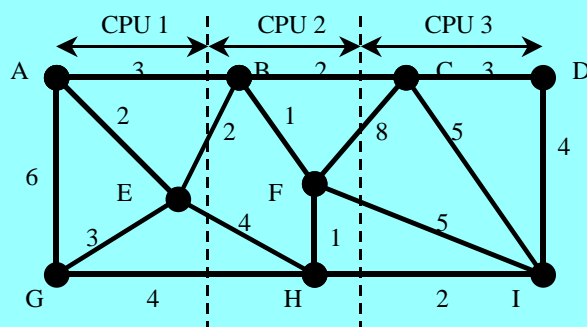
ASPECTOS DE LA IMPLANTACION DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v CASI TODOS LOS ALGORITMOS SUPONEN QUE LAS MAQUINAS CONOCEN SU PROPIA CARGA Y QUE PUEDEN INFORMAR SU ESTADO:
 - u LA *MEDICION DE LA CARGA* NO ES TAN SENCILLA.
 - u UN METODO CONSISTE EN CONTAR EL N° DE PROCESOS (HAY QUE CONSIDERAR LOS PROCESOS LATENTES NO ACTIVOS).
 - u OTRO METODO CONSISTE EN CONTAR SOLO LOS PROCESOS EN EJECUCION O LISTOS.
 - u TAMBIEN SE PUEDE MEDIR LA FRACCION DE TIEMPO QUE LA CPU ESTA OCUPADA.
- v OTRO ASPECTO IMPORTANTE ES EL *COSTO EXCESIVO* EN CONSUMO DE RECURSOS PARA RECOLECTAR MEDIDAS Y DESPLAZAR PROCESOS:
 - u SE DEBERIA CONSIDERAR EL TIEMPO DE CPU, EL USO DE MEMORIA Y EL ANCHO DE BANDA DE LA RED UTILIZADA POR EL ALGORITMO PARA ASIGNACION DE PROCESADORES.
- v SE DEBE CONSIDERAR LA *COMPLEJIDAD DEL SOFTWARE* EN CUESTION Y SUS IMPLICANCIAS PARA:
 - u EL DESEMPEÑO, LA CORRECTEZ Y LA ROBUSTEZ DEL SISTEMA.
- v SI EL USO DE UN ALGORITMO SENCILLO PROPORCIONA CASI LA MISMA GANANCIA QUE UNO MAS CARO Y MAS COMPLEJO:
 - u GENERALMENTE SERA MEJOR UTILIZAR EL MAS SENCILLO.
- v SE DEBE OTORGAR GRAN IMPORTANCIA A LA *ESTABILIDAD DEL SISTEMA*:
 - u LAS MAQUINAS EJECUTAN SUS ALGORITMOS EN FORMA ASINCRONA POR LO QUE EL SISTEMA NUNCA SE EQUILIBRA.
 - u LA MAYORIA DE LOS ALGORITMOS QUE INTERCAMBIAN INFORMACION:
 - Φ SON CORRECTOS LUEGO DE INTERCAMBIAR LA INFORMACION Y DE QUE TODO SE HA REGISTRADO.
 - Φ SON POCO CONFIABLES MIENTRAS LAS TABLAS CONTINUAN SU ACTUALIZACION:
 - SE PRESENTAN SITUACIONES DE NO EQUILIBRIO.

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v UN ALGORITMO DETERMINISTA SEGUN LA TEORIA DE GRAFICAS.
- v ES APLICABLE A SISTEMAS DONDE SE CONOCE:
 - v REQUERIMIENTOS DE CPU Y DE MEMORIA DE LOS PROCESOS.
 - v TRAFICO PROMEDIO ENTRE CADA PAR DE PROCESOS.
- v SI EL N° DE PROCESOS SUPERA AL N° DE CPU:
 - v HABRA QUE ASIGNAR VARIOS PROCESOS A LA MISMA CPU.
 - v LA ASIGNACION DEBERA *MINIMIZAR EL TRAFICO EN LA RED*.
- v EL SISTEMA SE PUEDE REPRESENTAR EN UNA GRAFICA CON PESOS:
 - v CADA NODO ES UN PROCESO.
 - v CADA ARCO ES EL FLUJO DE MENSAJES ENTRE DOS PROCESOS.
- v EL PROBLEMA ES ENCONTRAR LA FORMA DE PARTIR LA GRAFICA EN SUBGRAFICAS SUJETAS A RESTRICCIONES (EJ: DE CPU Y DE MEMORIA):
 - v LOS ARCOS QUE VAN DE UNA SUBGRAFICA A LA OTRA REPRESENTAN EL TRAFICO EN LA RED.
 - v CADA SUBGRAFICA ES UNA UNIDAD DE ASIGNACION.
 - v EL ALGORITMO DEBE BUSCAR UNIDADES DE ASIGNACION FUERTEMENTE ACOPLADAS:
 - Φ TRAFICO INTENSO DENTRO DE LA UNIDAD DE ASIGNACION.
 - Φ TRAFICO ESCASO ENTRE UNIDADES DE ASIGNACION.

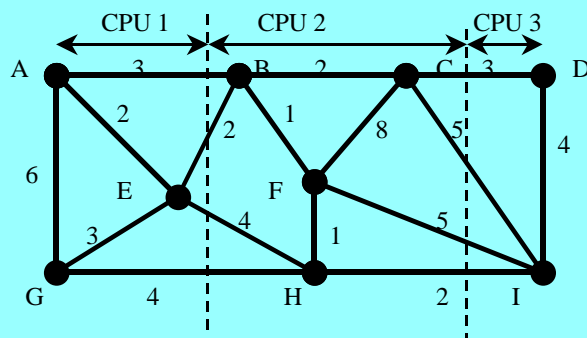
DOS FORMAS DE ASIGNAR 9 PROCESOS A 3 PROCESADORES



EL TRAFICO TOTAL EN LA RED ES LA SUMA DE LAS UNIDADES DE TRAFICO DE LOS ARCOS INTERSECTADOS POR LAS LINEAS PUNTEADAS:
 $3 + 2 + 4 + 4 + 2 + 8 + 5 + 2 = 30$ UNIDADES

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

DOS FORMAS DE ASIGNAR 9 PROCESOS A 3 PROCESADORES (CONTINUACION)



EL TRAFICO TOTAL EN LA RED ES LA SUMA DE LAS UNIDADES DE TRAFICO DE LOS ARCOS INTERSECTADOS POR LAS LINEAS PUNTEADAS:
 $3 + 2 + 4 + 4 + 3 + 5 + 5 + 2 = 28$ UNIDADES

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v **UN ALGORITMO CENTRALIZADO.**
- v ES UN ALGORITMO HEURISTICO QUE A DIFERENCIA DEL ANTERIOR NO PRECISA INFORMACION ANTICIPADAMENTE.
- v ES UN ALGORITMO **ARRIBA-ABAJO** (MUTKA Y LIVNY) **CENTRALIZADO** PORQUE UN COORDINADOR MANTIENE UNA *TABLA DE USOS*:
 - u CONTIENE UNA ENTRADA POR ESTACION DE TRABAJO INICIALIZADA EN “0”.
 - u CUANDO OCURREN EVENTOS SIGNIFICATIVOS SE ENVIAN AL COORDINADOR MENSAJES PARA ACTUALIZAR LA TABLA.
 - u LAS DECISIONES DE ASIGNACION SE BASAN EN LA TABLA:
 - Φ SE TOMAN CUANDO OCURREN EVENTOS DE PLANIFICACION:
 - SE REALIZA UNA SOLICITUD, SE LIBERA UN PROCESADOR, EL RELOJ PRODUCE UNA MARCA DE TIEMPO.
 - u NO SE INTENTA MAXIMIZAR EL USO DE LA CPU.
 - u SE PROCURA OTORGAR A CADA USUARIO UNA PARTE JUSTA DEL PODER DE COMPUTO.
 - u CUANDO LA MAQUINA DONDE SE CREA UN PROCESO DECIDE QUE SE DEBE EJECUTAR EN OTRA PARTE:
 - Φ LE PIDE AL COORDINADOR DE LA TABLA DE USOS QUE LE ASIGNE UN PROCESADOR:
 - SI EXISTE UNO DISPONIBLE Y NADIE MAS LO DESEA, SE OTORGA EL PERMISO.
 - SI NO, LA SOLICITUD SE NIEGA Y SE REGISTRA.
 - u SI UN USUARIO EJECUTA PROCESOS EN MAQUINAS DE OTROS USUARIOS ACUMULA PUNTOS DE PENALIZACION POR SEGUNDO:
 - Φ SE REGISTRA EN LA TABLA DE USOS.
 - u SI UN USUARIO TIENE SOLICITUDES PENDIENTES INSATISFECHAS:
 - Φ SE RESTAN PUNTOS DE PENALIZACION.
 - u SI NO EXISTEN SOLICITUDES PENDIENTES Y NINGUN PROCESADOR ESTA EN USO:
 - Φ LA ENTRADA DE LA TABLA DE USOS SE DESPLAZA UN CIERTO N° DE PUNTOS HACIA EL “0”, HASTA ALCANZARLO.
 - u EL MOVIMIENTO DE PUNTOS HACIA ARRIBA Y ABAJO DA NOMBRE AL ALGORITMO.

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v UN PUNTAJE POSITIVO EN UNA ENTRADA DE LA TABLA DE USOS INDICA QUE LA ESTACION DE TRABAJO RELACIONADA ES UN USUARIO DE LOS RECURSOS DEL SISTEMA.
- v UN PUNTAJE NEGATIVO SIGNIFICA QUE PRECISA RECURSOS.
- v UNA PUNTUACION "0" ES NEUTRA.
- v LA HEURISTICA UTILIZADA PARA LA ASIGNACION DE PROCESADORES ES LA SIGUIENTE:
 - u CUANDO UN PROCESADOR SE LIBERA GANA LA SOLICITUD PENDIENTE CUYO POSEEDOR TIENE LA PUNTUACION MENOR.
 - u UN USUARIO QUE NO OCUPE PROCESADORES Y QUE TENGA PENDIENTE UNA SOLICITUD DURANTE MUCHO TIEMPO:
 - Φ SIEMPRE VENCERA A ALGUIEN QUE UTILICE MUCHOS PROCESADORES.
 - Φ SE CUMPLE CON EL PRINCIPIO DE ASIGNAR LA CAPACIDAD DE MANERA JUSTA.
- v **UN ALGORITMO JERARQUICO.**
- v EL ALGORITMO ANTERIOR NO SE ADAPTA BIEN A LOS SISTEMAS DE GRAN TAMAÑO:
 - u EL NODO CENTRAL SE CONVIERTE EN UN CUELLO DE BOTELLA Y EN UN UNICO PUNTO DE FALLO.
- v UNA SOLUCION SON LOS ALGORITMOS JERARQUICOS QUE:
 - u MANTIENEN LA SENCILLEZ DE LOS CENTRALIZADOS.
 - u SE ESCALAN MEJOR QUE LOS CENTRALIZADOS.
- v UN METODO CONSISTE EN ORGANIZAR A LOS PROCESADORES EN JERARQUIAS LOGICAS INDEPENDIENTES DE LA ESTRUCTURA FISICA:
 - u SE ESTABLECE UN ARBOL JERARQUICO CON DISTINTOS NIVELES.
 - u PARA CADA GRUPO DE MAQUINAS HAY UNA MAQUINA ADMINISTRADORA:
 - Φ MANTIENE UN REGISTRO DE LAS MAQUINAS OCUPADAS Y LAS INACTIVAS.
 - u CADA PROCESADOR SE COMUNICA CON UN SUPERIOR Y UN N° REDUCIDO DE SUBORDINADOS:
 - Φ EL FLUJO DE INFORMACION ES CONTROLABLE.

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v EN CASO DE FALLA DE UN EQUIPO CON FUNCIONES JERARQUICAS:
 - u LO PUEDE REEMPLAZAR UN SUBORDINADO:
 - Φ LA ELECCION LA PUEDEN HACER LOS SUBORDINADOS, LOS PARES JERARQUICOS DEL EQUIPO FALLADO O EL SUPERIOR JERARQUICO DEL MISMO.
- v PARA DISMINUIR LA VULNERABILIDAD SE PUEDE TENER EN LA CIMA DEL ARBOL JERARQUICO NO UNO SINO UN GRUPO DE EQUIPOS:
 - u SI ALGUNO DEL GRUPO FALLA LOS RESTANTES ELIGEN A UN SUBORDINADO PARA INTEGRAR EL GRUPO SUPERIOR.
- v LAS TAREAS SE PUEDEN CREAR EN CUALQUIER PARTE DE LA JERARQUIA Y PUEDEN REQUERIR VARIOS PROCESOS, ES DECIR VARIOS PROCESADORES.
- v CADA ADMINISTRADOR DEBE MANTENER UN REGISTRO DE SUS EQUIPOS DEPENDIENTES QUE ESTEN DISPONIBLES.
- v SI EL ADMINISTRADOR QUE RECIBE UNA SOLICITUD DETERMINA QUE NO TIENE SUFICIENTES PROCESADORES DISPONIBLES:
 - u TRANSFIERE LA SOLICITUD HACIA ARRIBA A SU SUPERIOR, QUIEN TAMBIEN PODRIA TRASLADARLA HACIA ARRIBA NUEVAMENTE.
- v SI EL ADMINISTRADOR DETERMINA QUE SI PUEDE SATISFACER LA SOLICITUD:
 - u DIVIDE LA SOLICITUD EN PARTES Y LA DISTRIBUYE A LOS ADMINISTRADORES SUBORDINADOS A EL.
 - u LOS SUBORDINADOS REPITEN ESTA OPERACION HASTA LLEGAR AL NIVEL INFERIOR.
 - u LOS PROCESADORES SE SEÑALAN COMO “OCUPADOS” Y EL N° DE PROCESADORES ASIGNADOS SE INFORMA HACIA ARRIBA.
- v UN IMPORTANTE PROBLEMA CONSISTE EN QUE PODRIA HABER VARIAS SOLICITUDES EN DISTINTAS ETAPAS DEL ALGORITMO DE ASIGNACION:
 - u PUEDE CONDUCIR A ESTIMACIONES NO ACTUALIZADAS DEL N° DE PROCESADORES DISPONIBLES (TAMBIEN PUDIERON SALIR DE SERVICIO ALGUNOS DE LOS CONSIDERADOS DISPONIBLES).
 - u PODRIAN PRESENTARSE SITUACIONES DE COMPETENCIA, BLOQUEO, ETC. EN EL INTENTO DE ASIGNACION DE PROCESADORES.

EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES

- v **UN ALGORITMO DISTRIBUIDO HEURISTICO (EAGER).**
- v **AL CREARSE UN PROCESO:**
 - u LA MAQUINA DONDE SE ORIGINA ENVIA MENSAJES DE PRUEBA A UNA MAQUINA ELEGIDA AL AZAR:
 - Φ PREGUNTA SI SU CARGA ESTA POR DEBAJO DE CIERTO VALOR DE REFERENCIA.
 - u SI LA RESPUESTA ES POSITIVA EL PROCESO SE ENVIA A ESE LUGAR.
 - u SI NO, SE ELIGE OTRA MAQUINA PARA LA PRUEBA.
 - u LUEGO DE “N” PRUEBAS NEGATIVAS EL ALGORITMO TERMINA Y EL PROCESO SE EJECUTA EN LA MAQUINA DE ORIGEN.
- v **UN ALGORITMO DE REMATES.**
- v **UTILIZA UN MODELO ECONOMICO CON:**
 - u COMPRADORES Y VENDEDORES DE SERVICIOS.
 - u PRECIOS ESTABLECIDOS POR LA OFERTA Y LA DEMANDA.
- v **LOS PROCESOS DEBEN COMPRAR TIEMPO DE CPU.**
- v **CADA PROCESADOR ANUNCIA SU PRECIOS MEDIANTE UN ARCHIVO QUE TODOS PUEDEN LEER (ES EL PRECIO PAGADO POR EL ULTIMO CLIENTE).**
- v **LOS DISTINTOS PROCESADORES PUEDEN TENER DISTINTOS PRECIOS SEGUN SUS CARACTERISTICAS Y SERVICIOS.**
- v **CUANDO UN PROCESO DESEA INICIAR UN PROCESO HIJO:**
 - u VERIFICA SI ALGUIEN OFRECE EL SERVICIO QUE NECESITA.
 - u DETERMINA EL CONJUNTO DE PROCESADORES QUE PUEDEN PRESTAR SUS SERVICIOS.
 - u SELECCIONA EL MEJOR CANDIDATO SEGUN PRECIO, RAPIDEZ, RELACION PRECIO / DESEMPEÑO, TIPO DE APLICACION, ETC.
 - u GENERA UNA OFERTA Y LA ENVIA A SU PRIMER OPCION.
- v **LOS PROCESADORES:**
 - u REUNEN LAS OFERTAS RECIBIDAS Y ELIGEN UNA.
 - u INFORMAN A LOS GANADORES Y PERDEDORES.
 - u EJECUTAN LOS PROCESOS.
 - u ACTUALIZAN LOS PRECIOS.

PLANIFICACION EN SISTEMAS DISTRIBUIDOS

- v GENERALMENTE CADA PROCESADOR HACE SU PLANIFICACION LOCAL (SI TIENE VARIOS PROCESOS EN EJECUCION) INDEPENDIENTEMENTE DE LO QUE HACEN LOS OTROS PROCESADORES.
- v LA PLANIFICACION INDEPENDIENTE NO ES EFICIENTE CUANDO SE EJECUTAN EN DISTINTOS PROCESADORES UN GRUPO DE PROCESOS:
 - u RELACIONADOS ENTRE SI.
 - u CON UNA GRAN INTERACCION ENTRE LOS PROCESOS.
- v SE NECESITA UNA FORMA DE GARANTIZAR QUE LOS PROCESOS CON COMUNICACION FRECUENTE SE EJECUTEN DE MANERA SIMULTANEA.
- v EN MUCHOS CASOS UN GRUPO DE PROCESOS RELACIONADOS ENTRE SI INICIARAN JUNTOS.
- v LA COMUNICACION DENTRO DE LOS GRUPOS DEBE PREVALECER SOBRE LA COMUNICACION ENTRE LOS GRUPOS.
- v SE DEBE DISPONER DE UN N° DE PROCESADORES SUFICIENTE PARA SOPORTAR AL GRUPO DE MAYOR TAMAÑO.
- v CADA PROCESADOR SE MULTIPROGRAMA CON “N” ESPACIOS PARA LOS PROCESOS (MULTIPROGRAMACION DE NIVEL “N”).
- v EL **ALGORITMO DE OUSTERHOUT** UTILIZA EL CONCEPTO DE **COPLANIFICACION**:
 - u TOMA EN CUENTA LOS PATRONES DE COMUNICACION ENTRE LOS PROCESOS DURANTE LA PLANIFICACION.
 - u DEBE GARANTIZAR QUE TODOS LOS MIEMBROS DEL GRUPO SE EJECUTEN AL MISMO TIEMPO.
 - u SE EMPLEA UNA MATRIZ CONCEPTUAL DONDE:
 - Φ LAS FILAS SON ESPACIOS DE TIEMPO.
 - Φ LAS COLUMNAS SON LAS TABLAS DE PROCESOS DE LOS PROCESADORES.
 - u CADA PROCESADOR DEBE UTILIZAR UN ALGORITMO DE PLANIFICACION ROUND ROBIN:
 - Φ TODOS LOS PROCESADORES EJECUTAN EL PROCESO EN EL ESPACIO “0” DURANTE UN CIERTO PERIODO FIJO.
 - Φ TODOS LOS PROCESADORES EJECUTAN EL PROCESO EN EL ESPACIO “1” DURANTE UN CIERTO PERIODO FIJO, ETC.
 - u SE DEBEN MANTENER SINCRONIZADOS LOS INTERVALOS DE TIEMPO.
 - u TODOS LOS MIEMBROS DE UN GRUPO SE DEBEN COLOCAR EN EL MISMO N° DE ESPACIO DE TIEMPO PERO EN PROCESADORES DISTINTOS.

SISTEMAS DISTRIBUIDOS DE ARCHIVOS

- v INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS
- v DISEÑO DE LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS
- v LA INTERFAZ DEL SERVICIO DE ARCHIVOS
- v LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS
- v SEMANTICA DE LOS ARCHIVOS COMPARTIDOS
- v IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS
- v USO DE ARCHIVOS
- v ESTRUCTURA DEL SISTEMA
- v OCULTAMIENTO
- v REPLICA
- v CONCLUSIONES IMPORTANTES RESPECTO DE LA IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS
- v TENDENCIAS EN LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS
- v CONSIDERACIONES RESPECTO DEL HARDWARE
- v ESCALABILIDAD
- v REDES EN UN AREA AMPLIA
- v USUARIOS MOVILES
- v TOLERANCIA DE FALLOS

INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS

- v MUCHOS ASPECTOS SON SIMILARES A LOS DE LOS SISTEMAS CONVENCIONALES CENTRALIZADOS.
- v EN UN SISTEMA DISTRIBUIDO ES IMPORTANTE DISTINGUIR ENTRE LOS CONCEPTOS DE *SERVICIO DE ARCHIVOS* Y EL *SERVIDOR DE ARCHIVOS*.
- v **EL SERVICIO DE ARCHIVOS:**
 - u ES LA ESPECIFICACION DE LOS SERVICIOS QUE EL SISTEMA DE ARCHIVOS OFRECE A SUS CLIENTES.
 - u DESCRIBE LAS PRIMITIVAS DISPONIBLES, LOS PARAMETROS QUE UTILIZAN Y LAS ACCIONES QUE LLEVAN A CABO.
 - u DEFINE PRECISAMENTE EL SERVICIO CON QUE PUEDEN CONTAR LOS CLIENTES SIN DECIR NADA RESPECTO DE SU IMPLANTACION.
- v **EL DESPACHADOR (SERVIDOR) DE ARCHIVOS:**
 - u ES UN PROCESO QUE SE EJECUTA EN ALGUNA MAQUINA Y AYUDA CON LA IMPLANTACION DEL SERVICIO DE ARCHIVOS.
 - u PUEDE HABER UNO O VARIOS EN UN SISTEMA.
 - u LOS CLIENTES NO DEBEN SER CONSCIENTES DE LA FORMA DE IMPLANTAR EL SISTEMA DE ARCHIVOS:
 - Φ NO PRECISAN CONOCER EL N° DE SERVIDORES DE ARCHIVOS, SU POSICION O FUNCION.
 - Φ DEBERIAN VER AL SISTEMA DISTRIBUIDO DE ARCHIVOS COMO UN SISTEMA DE ARCHIVOS NORMAL DE UNIPROCESADOR.
- v GENERALMENTE UN SERVIDOR DE ARCHIVOS ES UN PROCESO DEL USUARIO (A VECES DEL NUCLEO) QUE SE EJECUTA EN UNA MAQUINA:
 - u UN SISTEMA PUEDE CONTENER VARIOS SERVIDORES DE ARCHIVOS, C / U CON UN SERVICIO DISTINTO:
 - Φ EJ: UN SISTEMA CON UN SERVIDOR DE ARCHIVOS EN “UNIX” Y OTRO EN “DOS”.
 - Φ CADA PROCESO USUARIO UTILIZARIA EL SERVIDOR APROPIADO.

DISEÑO DE LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS

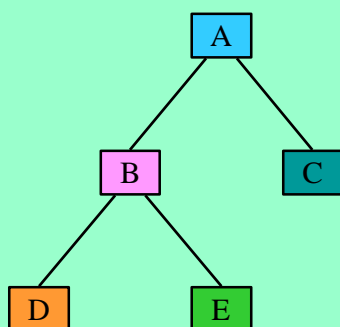
LA INTERFAZ DEL SERVICIO DE ARCHIVOS

- v **DISEÑO DE LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS.**
- v LOS COMPONENTES DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS SON:
 - u EL VERDADERO SERVICIO DE ARCHIVOS:
 - Φ REALIZA OPERACIONES EN LOS ARCHIVOS INDIVIDUALES:
 - LECTURA, ESCRITURA, ADICION.
 - u EL SERVICIO DE DIRECTORIOS:
 - Φ CREA Y MANEJA DIRECTORIOS, AÑADE Y ELIMINA ARCHIVOS DE LOS DIRECTORIOS, ETC.
- v **LA INTERFAZ DEL SERVICIO DE ARCHIVOS.**
- v LA PROTECCION EN LOS SISTEMAS DISTRIBUIDOS UTILIZA LAS MISMAS TECNICAS DE LOS SISTEMAS CON UNIPROCESADOR:
 - u **POSIBILIDADES:**
 - Φ CADA USUARIO TIENE UN PERMISO O **POSIBILIDAD** PARA CADA OBJETO AL QUE TIENE ACCESO:
 - DETERMINA LOS TIPOS DE ACCESOS PERMITIDOS.
 - u **LISTAS PARA CONTROL DE ACCESO:**
 - Φ SE ASOCIA A CADA ARCHIVO UNA LISTA IMPLICITA O EXPLICITA DE:
 - LOS USUARIOS QUE PUEDEN TENER ACCESO AL ARCHIVO.
 - LOS TIPOS DE ACCESO PERMITIDOS A C / U DE ELLOS.
- v LOS SERVICIOS DE ARCHIVOS SE PUEDEN CLASIFICAR EN DOS TIPOS:
 - u **MODELO CARGA / DESCARGA:**
 - Φ LAS PRINCIPALES OPERACIONES SON LA LECTURA DE UN ARCHIVO Y LA ESCRITURA EN UN ARCHIVO.
 - Φ LA LECTURA TRANSFIERE *TODO* UN ARCHIVO DE UNO DE LOS SERVIDORES DE ARCHIVOS AL CLIENTE SOLICITANTE.
 - Φ LA ESCRITURA TRANSFIERE EN SENTIDO CONTRARIO.
 - Φ LOS ARCHIVOS SE PUEDEN ALMACENAR EN MEMORIA O EN UN DISCO LOCAL.
 - u **MODELO DE ACCESO REMOTO:**
 - Φ EL SISTEMA DE ARCHIVOS SE EJECUTA CON TODAS LAS FUNCIONES EN LOS SERVIDORES Y NO EN LOS CLIENTES.

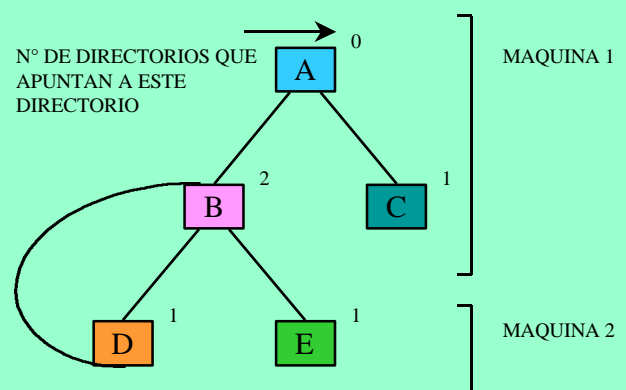
LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS

- v PROPORCIONA OPERACIONES PARA CREAR Y ELIMINAR DIRECTORIOS, NOMBRAR Y CAMBIAR EL NOMBRE DE ARCHIVOS Y MOVER ARCHIVOS DE UN DIRECTORIO A OTRO.
- v SE UTILIZA UN **SISTEMA JERARQUICO DE ARCHIVOS**, REPRESENTADO POR UN ARBOL DE DIRECTORIOS.
- v EN CIERTOS SISTEMAS ES POSIBLE CREAR ENLACES O APUNTADES A UN DIRECTORIO ARBITRARIO:
 - v SE PUEDEN COLOCAR EN CUALQUIER DIRECTORIO.
 - v SE PUEDEN CONSTRUIR GRAFICAS DE DIRECTORIOS.
- v EN UNA JERARQUIA CON ESTRUCTURA DE ARBOL SOLO SE PUEDE ELIMINAR UN ENLACE CON UN DIRECTORIO SI EL DIRECTORIO AL CUAL SE APUNTA ESTA VACIO.
- v EN UNA GRAFICA SE PERMITE LA ELIMINACION DE UN ENLACE MIENTRAS EXISTA AL MENOS OTRO:
 - v SE UTILIZA UN CONTADOR DE REFERENCIAS PARA DETERMINAR SI EL ENLACE POR ELIMINAR ES EL ULTIMO.
 - v SE PUEDE ARMAR UNA GRAFICA DE DIRECTORIOS COMPRENDIENDO A DIRECTORIOS DE DOS O MAS MAQUINAS.
 - v LA ELIMINACION DE ENLACES PUEDE LLEVAR A DIRECTORIOS Y ARCHIVOS A LA CONDICION DE HUERFANOS:
 - ❖ NO PUEDEN SER ALCANZADOS DESDE EL DIRECTORIO RAIZ.

**ARBOL DE DIRECTORIOS
CONTENIDO EN UNA MAQUINA**



**GRAFICA DE DIRECTORIOS
EN DOS MAQUINAS**



LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS

- v UN ASPECTO FUNDAMENTAL DE DISEÑO EN SISTEMAS DISTRIBUIDOS ES SI TODAS LAS MAQUINAS Y PROCESOS TENDRAN EXACTAMENTE LA MISMA *VISION DE LA JERARQUIA DE LOS DIRECTORIOS*.
- v EN LOS SISTEMAS QUE UTILIZAN VARIOS SERVIDORES DE ARCHIVOS MEDIANTE EL MONTAJE REMOTO GENERALMENTE LOS DIVERSOS CLIENTES TIENEN UNA VISION DIFERENTE DEL SISTEMA DE ARCHIVOS:
 - u LA DESVENTAJA ES QUE EL SISTEMA NO SE COMPORTA COMO UN UNICO SISTEMA DE TIEMPO COMPARTIDO.
- v UNA CUESTION RELACIONADA ES SI EXISTE UN DIRECTORIO RAIZ GLOBAL AL QUE TODAS LAS MAQUINAS RECONOZCAN COMO LA RAIZ:
 - u UNA POSIBILIDAD ES QUE LA RAIZ SOLO CONTENGA UNA ENTRADA POR CADA SERVIDOR.
- v **TRANSPARENCIA DE LOS NOMBRES.**
- v LA **TRANSPARENCIA CON RESPECTO A LA POSICION** SIGNIFICA QUE EL NOMBRE DE LA RUTA DE ACCESO NO SUGIERE LA POSICION DEL ARCHIVO:
 - u SE INDIVIDUALIZA AL SERVIDOR PERO NO SE INDICA DONDE ESTA:
 - Φ PUEDE MOVERSE DENTRO DE LA RED SIN NECESIDAD DE CAMBIAR LA RUTA.
 - u EJ.: /servidor1/dir1/dir2/x.
- v SI EL PRIMER COMPONENTE DE TODAS LAS RUTAS DE ACCESO ES EL SERVIDOR:
 - u EL SISTEMA NO PUEDE DESPLAZAR EL ARCHIVO A OTRO SERVIDOR EN FORMA AUTOMATICA PORQUE CAMBIARIA EL NOMBRE DE LA RUTA DE ACCESO.
- v UN SISTEMA DONDE LOS ARCHIVOS SE PUEDEN DESPLAZAR SIN QUE CAMBIEN SUS NOMBRES TIENE **INDEPENDENCIA CON RESPECTO A LA POSICION**.
- v RESUMIENDO, LOS METODOS USUALES PARA NOMBRAR LOS ARCHIVOS Y DIRECTORIOS EN UN SISTEMA DISTRIBUIDO SON:
 - u NOMBRE MAQUINA + RUTA DE ACCESO.
 - u MONTAJE DE SISTEMAS DE ARCHIVOS REMOTOS EN LA JERARQUIA LOCAL DE ARCHIVOS.
 - u UN UNICO ESPACIO DE NOMBRES QUE TENGA LA MISMA APARIENCIA EN TODAS LAS MAQUINAS.

SEMANTICA DE LOS ARCHIVOS COMPARTIDOS

- v CUANDO SE COMPARTEN ARCHIVOS ES NECESARIO DEFINIR CON PRECISION LA SEMANTICA DE LA LECTURA Y ESCRITURA.
- v EN SISTEMAS MONOPROCESADOR QUE PERMITEN A LOS PROCESOS COMPARTIR ARCHIVOS (EJ.: UNIX) LA SEMANTICA GENERALMENTE ESTABLECE:
 - v SI UN *READ* SIGUE A UN *WRITE*, *READ* DEBE REGRESAR EL VALOR RECIEN ESCRITO.
 - v SI DOS *WRITE* SE REALIZAN EN SERIE Y LUEGO SE EJECUTA UN *READ*, EL VALOR QUE SE DEBE REGRESAR ES EL ALMACENADO EN LA ULTIMA ESCRITURA.
 - v ESTE MODELO SE DENOMINA **SEMANTICA DE UNIX**.
- v EN UN SISTEMA DISTRIBUIDO LA SEMANTICA DE UNIX SE PUEDE LOGRAR FACILMENTE SI:
 - v SOLO EXISTE UN SERVIDOR DE ARCHIVOS.
 - v LOS CLIENTES NO OCULTAN LOS ARCHIVOS.
- v UN PROBLEMA QUE SE PUEDE PRESENTAR SE DEBE A LOS RETRASOS EN LA RED:
 - v SI UN *READ* OCURRIDO DESPUES DE UN *WRITE* LLEGA PRIMERO AL SERVIDOR OBTENDRA EL VALOR PREVIO AL *WRITE*.
- v OTRO PROBLEMA ES EL DESEMPEÑO POBRE DE UN SISTEMA DISTRIBUIDO EN DONDE TODAS LAS SOLICITUDES DE ARCHIVOS DEBEN PASAR A UN UNICO SERVIDOR:
 - v UNA SOLUCION ES PERMITIR A LOS CLIENTE MANTENER COPIAS LOCALES DE LOS ARCHIVOS DE USO FRECUENTE EN SUS CACHES PARTICULARES, LO QUE OCASIONA EL SIGUIENTE PROBLEMA:
 - ⦿ UN CLIENTE MODIFICA LOCALMENTE UN ARCHIVO EN SU CACHE.
 - ⦿ LUEGO OTRO CLIENTE LEE EL ARCHIVO DEL SERVIDOR.
 - ⦿ EL SEGUNDO CLIENTE OBTENDRA UN ARCHIVO OBSOLETO.
 - v UNA SOLUCION SERIA PROPAGAR INMEDIATAMENTE TODAS LAS MODIFICACIONES DE LOS ARCHIVOS EN CACHE DE REGRESO AL DESPACHADOR:
 - ⦿ RESULTA PRACTICAMENTE INEFICIENTE.

SEMANTICA DE LOS ARCHIVOS COMPARTIDOS

- v OTRA SOLUCION ES RELAJAR LA SEMANTICA DE LOS ARCHIVOS COMPARTIDOS:
 - υ LOS CAMBIOS A UN ARCHIVO ABIERTO SOLO PUEDEN SER VISTOS EN UN PRINCIPIO POR EL PROCESO (O TAL MAQUINA) QUE MODIFICO EL ARCHIVO.
 - υ LOS CAMBIOS SERAN VISIBLES A LOS DEMAS PROCESOS (O MAQUINAS) SOLO CUANDO SE CIERRE EL ARCHIVO Y SEA ACTUALIZADO EN EL SERVIDOR.
 - υ ESTA REGLA SE CONOCE COMO LA **SEMANTICA DE SESION**.
- v UN PROBLEMA SE PRESENTA CUANDO DOS O MAS CLIENTES OCULTAN Y MODIFICAN EL MISMO ARCHIVO EN FORMA SIMULTANEA:
 - υ UNA SOLUCION ES QUE AL CERRAR CADA ARCHIVO SU VALOR SE ENVIA DE REGRESO AL SERVIDOR:
 - ⊕ EL RESULTADO FINAL DEPENDE DE QUIEN LO CIERRE MAS RAPIDO.
- v OTRO PROBLEMA CONSISTE EN QUE NO SE PUEDEN COMPARTIR LOS APUNTADES QUE PARA CADA ARCHIVO INDICAN EN LA SEMANTICA UNIX LA POSICION ACTUAL EN EL ARCHIVO.
- v UN METODO DISTINTO ES QUE **TODOS LOS ARCHIVOS SEAN INMUTABLES**:
 - υ NO SE PUEDE ABRIR UN ARCHIVO PARA ESCRIBIR EN EL.
 - υ SOLO SE PERMITEN LAS OPERACIONES *CREATE* Y *READ*.
 - υ LOS DIRECTORIOS SI SE PUEDEN ACTUALIZAR.
 - υ SE PUEDE CREAR UN ARCHIVO NUEVO E INTRODUCIRLO EN EL DIRECTORIO CON EL NOMBRE DE UN ARCHIVO YA EXISTENTE:
 - ⊕ ESTE SE VUELVE INACCESIBLE CON EL MISMO NOMBRE.
 - ⊕ PERSISTE EL PROBLEMA DE COMO TRATAR LA SITUACION PRESENTADA CUANDO DOS PROCESOS INTENTAN REEMPLAZAR EL MISMO ARCHIVO A LA VEZ.
- v OTRA VIA DE SOLUCION PARA EL USO DE ARCHIVOS COMPARTIDOS EN UN SISTEMA DISTRIBUIDO ES USAR LAS **TRANSACCIONES ATOMICAS**:
 - υ SE GARANTIZA QUE TODAS LAS LLAMADAS CONTENIDAS EN LA TRANSACCION SE LLEVARAN A CABO EN ORDEN.
 - υ NO HABRA INTERFERENCIAS DE OTRAS TRANSACCIONES CONCURRENTES.

IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS

USO DE ARCHIVOS

- v **IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS.**
- v LA IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS INCLUYE ASPECTOS TALES COMO:
 - υ EL USO DE LOS ARCHIVOS.
 - υ LA ESTRUCTURA DEL SISTEMA.
 - υ EL OCULTAMIENTO.
 - υ LA DUPLICACION O REPLICA.
 - υ EL CONTROL DE LA CONCURRENCIA.
- v **USO DE ARCHIVOS.**
- v ANTES DE IMPLANTAR UN SISTEMA DE ARCHIVOS RESULTA DE INTERES ANALIZAR LOS “*PATRONES DE USO*” DE DICHOS ARCHIVOS.
- v PARA DETERMINAR LOS PATRONES DE USO ES NECESARIO TOMAR MEDICIONES QUE PUEDEN SER:
 - υ ESTATICAS:
 - ⊕ REPRESENTAN UNA TOMA INSTANTANEA DEL SISTEMA EN UN MOMENTO DADO.
 - ⊕ COMPRENDEN LA DISTRIBUCION DE TAMAÑO DE LOS ARCHIVOS, LA DISTRIBUCION DE TIPO DE ARCHIVOS, LA CANTIDAD DE ESPACIO QUE OCUPAN LOS ARCHIVOS DE VARIOS TAMAÑOS Y TIPOS, ETC.
 - υ DINAMICAS:
 - ⊕ REGISTRAN EN UNA BITACORA TODAS LAS OPERACIONES QUE MODIFICAN EL SISTEMA DE ARCHIVOS.
 - ⊕ COMPRENDEN INFORMACION SOBRE LA FRECUENCIA RELATIVA DE VARIAS OPERACIONES, EL N° DE ARCHIVOS ABIERTOS EN UN MOMENTO DADO, LA CANTIDAD DE ARCHIVOS COMPARTIDOS, ETC.
- v LAS PRINCIPALES PROPIEDADES OBSERVADAS SON:
 - υ LA MAYORIA DE LOS ARCHIVOS SON PEQUEÑOS.
 - υ LA LECTURA ES MAS COMUN QUE LA ESCRITURA.
 - υ LA MAYORIA DE LOS ACCESOS ES SECUENCIAS.
 - υ LA MAYORIA DE LOS ARCHIVOS SON DE CORTA VIDA.
 - υ ES POCO USUAL COMPARTIR ARCHIVOS.
 - υ LOS PROCESOS PROMEDIO UTILIZAN POCOS ARCHIVOS.
 - υ DISTINTAS CLASES DE ARCHIVOS POSEEN PROPIEDADES DISTINTAS.

ESTRUCTURA DEL SISTEMA

- v EN CIERTOS SISTEMAS NO EXISTE DISTINCION ENTRE UN CLIENTE Y UN SERVIDOR:
 - u TODAS LAS MAQUINAS EJECUTAN EL MISMO SOFTWARE BASICO.
 - u UNA MAQUINA QUE DESEE DAR SERVICIO DE ARCHIVOS LO PUEDE HACER:
 - Φ DEBE EXPORTAR LOS NOMBRES DE LOS DIRECTORIOS SELECCIONADOS, PARA QUE OTRAS MAQUINAS LOS PUEDAN ACCEDER.
- v EN OTROS SISTEMAS EL SERVIDOR DE ARCHIVOS Y EL DE DIRECTORIOS SON SOLO PROGRAMAS DEL USUARIO:
 - u SE PUEDE CONFIGURAR UN SISTEMA PARA QUE EJECUTE O NO EL SOFTWARE DE CLIENTE O SERVIDOR EN LA MISMA MAQUINA.
- v LOS CLIENTES Y SERVIDORES TAMBIEN PODRIAN SER MAQUINAS TOTALMENTE DISTINTAS EN TERMINOS DE HARDWARE O DE SOFTWARE.
- v UN ASPECTO DE IMPLANTACION EN DONDE DIFIEREN LOS SISTEMAS ES LA FORMA DE ESTRUCTURAR EL SERVICIO A DIRECTORIOS Y ARCHIVOS; LAS PRINCIPALES OPCIONES SON LAS SIGUIENTES:
 - u COMBINAR EL SERVICIO A DIRECTORIOS Y ARCHIVOS EN UN UNICO SERVIDOR QUE ADMINISTRE TODAS LAS LLAMADAS:
 - Φ A DIRECTORIOS Y ARCHIVOS.
 - u SEPARAR EL SERVICIO A DIRECTORIOS Y ARCHIVOS UTILIZANDO UN SERVIDOR DE DIRECTORIOS Y UN SERVIDOR DE ARCHIVOS.
- v SI SE CONSIDERA EL CASO DE SERVIDORES DE ARCHIVOS Y DIRECTORIOS INDEPENDIENTES:
 - u EL CLIENTE ENVIA UN NOMBRE SIMBOLICO AL SERVIDOR DE DIRECTORIOS.
 - u EL SERVIDOR DE DIRECTORIOS REGRESA EL NOMBRE EN BINARIO (EJ: MAQUINA + NODO_I) QUE COMPRENDE EL SERVIDOR DE ARCHIVOS.
 - u ES POSIBLE QUE UNA JERARQUIA DE DIRECTORIOS SE REPARTA ENTRE VARIOS SERVIDORES.
 - u EL SERVIDOR QUE RECIBE UN NOMBRE BINARIO QUE SE REFIERE A OTRO SERVIDOR PUEDE:
 - Φ INDICAR AL CLIENTE EL SERVIDOR QUE TIENE EL ARCHIVO BUSCADO, PARA QUE EL CLIENTE LO BUSQUE.
 - Φ ENVIAR LA SOLICITUD AL SIGUIENTE SERVIDOR Y NO CONTESTAR.

ESTRUCTURA DEL SISTEMA

- v UN ASPECTO ESTRUCTURAL A CONSIDERAR ES SI LOS SERVIDORES DE ARCHIVOS, DIRECTORIOS O DE OTRO TIPO DEBEN CONTENER LA *INFORMACION DE ESTADO DE LOS CLIENTES*.
- v UNA POSIBILIDAD ES QUE LOS SERVIDORES NO DEBEN CONTENER LOS ESTADOS, DEBEN SER **SIN ESTADO**:
 - u CUANDO UN CLIENTE ENVIA UNA SOLICITUD A UN SERVIDOR:
 - Φ EL SERVIDOR LA LLEVA A CABO, ENVIA LA RESPUESTA Y ELIMINA DE SUS TABLAS INTERNAS TODA LA INFORMACION RELATIVA A ESA SOLICITUD.
 - Φ EL SERVIDOR NO GUARDA INFORMACION RELATIVA A LOS CLIENTES ENTRE LAS SOLICITUDES.
- v OTRA POSIBILIDAD ES QUE LOS SERVIDORES **CONSERVEN INFORMACION DE ESTADO** DE LOS CLIENTES ENTRE LAS SOLICITUDES.
- v ACLARACION:
 - u LUEGO DE ABRIR UN ARCHIVO EL SERVIDOR DEBE MANTENER LA INFORMACION QUE RELACIONE LOS CLIENTES CON LOS ARCHIVOS ABIERTOS POR ESTOS.
 - u AL ABRIR UN ARCHIVO EL CLIENTE RECIBE UN DESCRIPTOR DE ARCHIVO QUE SE UTILIZA EN LAS LLAMADAS POSTERIORES PARA IDENTIFICACION DEL ARCHIVO.
 - u AL RECIBIR UNA SOLICITUD EL SERVIDOR UTILIZA EL DESCRIPTOR DE ARCHIVO PARA DETERMINAR EL ARCHIVO NECESARIO.
 - u LA TABLA QUE ASOCIA LOS DESCRIPTORES DE ARCHIVO CON LOS ARCHIVOS PROPIAMENTE DICHOS ES *INFORMACION DE ESTADO*.
- v EN UN SERVIDOR SIN ESTADO CADA SOLICITUD DEBE SER AUTOCONTENIDA:
 - u DEBE INCLUIR EL NOMBRE DEL ARCHIVO Y TODA LA INFORMACION PARA QUE EL SERVIDOR REALICE EL TRABAJO.
 - u LA LONGITUD DEL MENSAJE ES MAYOR.
- v SI UN SERVIDOR CON ESTADO FALLA Y SUS TABLAS SE PIERDEN:
 - u AL VOLVER A ARRANCAR NO TIENE INFORMACION SOBRE LA RELACION ENTRE LOS CLIENTES Y LOS ARCHIVOS ABIERTOS POR ESTOS.
 - u LA RECUPERACION QUEDA A CARGO DE LOS CLIENTES.
- v LOS SERVIDORES SIN ESTADO SON TIENDEN A SER MAS TOLERANTES DE LOS FALLOS QUE LOS SERVIDORES CON ESTADOS.

OCULTAMIENTO

- v EN UN SISTEMA CLIENTE - SERVIDOR, C / U CON SU MEMORIA PRINCIPAL Y UN DISCO, EXISTEN CUATRO LUGARES DONDE SE PUEDEN ALMACENAR LOS ARCHIVOS O PARTES DE ELLOS:
 - v EL DISCO DEL SERVIDOR.
 - v LA MEMORIA PRINCIPAL DEL SERVIDOR.
 - v EL DISCO DEL CLIENTE (SI EXISTE).
 - v LA MEMORIA PRINCIPAL DEL CLIENTE.
- v SI LOS ARCHIVOS SE ALMACENAN EN EL DISCO DEL SERVIDOR:
 - v DISPONEN DE ABUNDANTE ESPACIO.
 - v SERIAN ACCESIBLES A TODOS LOS CLIENTES.
 - v NO HABRA PROBLEMAS DE CONSISTENCIA AL EXISTIR SOLO UNA COPIA DE CADA ARCHIVO.
 - v PUEDE HABER PROBLEMAS DE DESEMPEÑO:
 - ⊕ ANTES DE QUE UN CLIENTE PUEDA LEER UN ARCHIVO SE LO DEBE TRANSFERIR:
 - DEL DISCO DEL SERVIDOR A LA MEMORIA PRINCIPAL DEL SERVIDOR.
 - DE LA MEMORIA PRINCIPAL DEL SERVIDOR A LA MEMORIA PRINCIPAL DEL CLIENTE, A TRAVES DE LA RED.
 - v SE PUEDE MEJORAR EL DESEMPEÑO **OCULTANDO (CONSERVANDO)** LOS ARCHIVOS DE MAS RECIENTE USO EN LA MEMORIA PRINCIPAL DEL SERVIDOR:
 - ⊕ UN CLIENTE QUE LEA UN ARCHIVO YA PRESENTE EN EL CACHE DEL SERVIDOR ELIMINA LA TRANSFERENCIA DEL DISCO.
 - ⊕ SE NECESITA UN ALGORITMO PARA DETERMINAR LOS ARCHIVOS O PARTES DE ARCHIVOS QUE DEBEN PERMANECER EN EL CACHE.
- v EL ALGORITMO DEBE RESOLVER LOS SIGUIENTES PROBLEMAS:
 - v LA UNIDAD QUE MANEJA EL CACHE.
 - v QUE HACER SI SE UTILIZA TODA LA CAPACIDAD DEL CACHE Y HAY QUE ELIMINAR A ALGUIEN.

OCULTAMIENTO

- v RESPECTO DE LA *UNIDAD QUE MANEJA EL CACHE*:
 - u PUEDE MANEJAR *ARCHIVOS COMPLETOS O BLOQUES DEL DISCO*.
 - u EL OCULTAMIENTO DE ARCHIVOS COMPLETOS QUE SE PUEDEN ALMACENAR EN FORMA ADYACENTE EN EL DISCO PERMITE UN BUEN DESEMPEÑO EN GENERAL.
 - u EL OCULTAMIENTO DE BLOQUES DE DISCO UTILIZA EL CACHE Y EL ESPACIO EN DISCO MAS EFICIENTEMENTE.
- v RESPECTO DE QUE HACER CUANDO *SE UTILIZA TODA LA CAPACIDAD DEL CACHE Y HAY QUE ELIMINAR A ALGUIEN*:
 - u SE PUEDE UTILIZAR CUALQUIER ALGORITMO DE OCULTAMIENTO, POR EJ.: LRU MEDIANTE LISTAS LIGADAS.
 - u CUANDO HAY QUE ELIMINAR A ALGUIEN DE LA MEMORIA:
 - Φ SE ELIGE AL MAS ANTIGUO.
 - Φ SI EXISTE UNA COPIA ACTUALIZADA EN EL DISCO SE DESCARTA LA COPIA DEL CACHE.
 - Φ DE LO CONTRARIO PRIMERO SE ACTUALIZA EL DISCO.
- v EL MANTENIMIENTO DE UN CACHE EN LA MEMORIA PRINCIPAL DEL SERVIDOR ES FACIL DE LOGRAR Y ES TOTALMENTE TRANSPARENTE A LOS CLIENTES.
- v SI SE UTILIZA OCULTAMIENTO EN EL LADO DEL CLIENTE:
 - u SE ELIMINA EL ACCESO A LA RED PARA TRANSFERIR DEL SERVIDOR AL CLIENTE.
 - u EL DISCO DEL CLIENTE GENERALMENTE ES MAS LENTO Y DE MENOR CAPACIDAD.
 - u GENERALMENTE ES MAS RAPIDO Y MAS SENCILLO TENER UN CACHE EN LA MEMORIA PRINCIPAL DEL SERVIDOR QUE EN EL DISCO DEL CLIENTE.
- v SI EL CACHE SE COLOCA EN LA MEMORIA PRINCIPAL DEL CLIENTE LAS PRINCIPALES OPCIONES SON:
 - u OCULTAR LOS ARCHIVOS DENTRO DEL PROPIO ESPACIO DE DIRECCIONES DE UN PROCESO DE USUARIO.
 - u COLOCAR EL CACHE EN EL NUCLEO.
 - u OCULTAR EL CACHE EN UN PROCESO MANEJADOR DEL CACHE, INDEPENDIENTE Y A NIVEL USUARIO.

OCULTAMIENTO

- v **CONSISTENCIA DEL CACHE.**
- v EL OCULTAMIENTO POR PARTE DEL CLIENTE INTRODUCE INCONSISTENCIA EN EL SISTEMA.
- v SI DOS CLIENTES LEEN UN MISMO ARCHIVO EN FORMA SIMULTANEA Y DESPUES LO MODIFICAN, APARECEN ALGUNOS PROBLEMAS:
 - u CUANDO UN TERCER PROCESO LEE EL ARCHIVO DEL SERVIDOR OBTENDRA LA VERSION ORIGINAL Y NO ALGUNA DE LAS NUEVAS:
 - ⊕ SE PUEDE EVITAR MEDIANTE LA “SEMANTICA DE SESION”:
 - NO ES APLICABLE CUANDO SE REQUIERE LA “SEMANTICA DE UNIX”.
 - u CUANDO DOS ARCHIVOS SE ESCRIBEN DE NUEVO AL SERVIDOR, EL ULTIMO DE ELLOS SE ESCRIBIRA SOBRE EL OTRO.
- v UNA SOLUCION A LA INCONSISTENCIA DEL CACHE ES EL ALGORITMO DE **ESCRITURA A TRAVES DEL CACHE**:
 - u CUANDO SE MODIFICA UNA ENTRADA DEL CACHE (ARCHIVO O BLOQUE), EL NUEVO VALOR:
 - ⊕ SE MANTIENE DENTRO DE EL.
 - ⊕ SE ENVIA DE INMEDIATO AL SERVIDOR.
- v LOS PRINCIPALES PROBLEMAS DE LA ESCRITURA A TRAVES DEL CACHE SON LOS SIGUIENTES:
 - u POSIBLE SUMINISTRO DE VALORES OBSOLETOS:
 - ⊕ UN PROCESO CLIENTE EN LA MAQUINA “A” LEE UN ARCHIVO “f” Y MANTIENE A “f” EN SU CACHE.
 - ⊕ UN CLIENTE EN LA MAQUINA “B” LEE EL MISMO ARCHIVO, LO MODIFICA Y LO ESCRIBE EN EL SERVIDOR.
 - ⊕ OTRO PROCESO CLIENTE INICIA EN LA MAQUINA “A” ABRIENDO Y LEYENDO “f”, QUE SE TOMA DEL CACHE.
 - ⊕ EL VALOR DE “f” ES OBSOLETO.
 - ⊕ UNA SOLUCION CONSISTE EN EXIGIR AL MANEJADOR DEL CACHE QUE VERIFIQUE EL SERVIDOR ANTES DE PROPORCIONAR AL CLIENTE UN ARCHIVO DEL CACHE:
 - GENERALMENTE UTILIZARA UNA “RPC” Y POCA INFORMACION DE CONTROL.

OCULTAMIENTO

- υ EL TRAFICO EN LA RED EN EL CASO DE LAS ESCRITURAS ES IGUAL QUE EN EL CASO DE NO OCULTAMIENTO:
 - ⊕ PARA MEJORAR SE PUEDE APLICAR EL SIGUIENTE PROCEDIMIENTO DE **RETRASO EN LA ESCRITURA**:
 - EN VEZ DE IR HACIA EL SERVIDOR EN EL INSTANTE EN QUE SE REALIZA LA ESCRITURA, EL CLIENTE:
 - HACE UNA NOTIFICACION DE QUE HA ACTUALIZADO UN ARCHIVO.
 - CADA CIERTO INTERVALO (EJ.: 30") TODAS LAS ACTUALIZACIONES SE AGRUPAN Y ENVIAN LA SERVIDOR AL MISMO TIEMPO (UN BLOQUE).
 - ⊕ EL RETRASO EN LA ESCRITURA OBSCURECE LA SEMANTICA:
 - SI OTRO PROCESO LEE EL ARCHIVO, EL RESULTADO DEPENDERA DE LA SINCRONIZACION DE LOS EVENTOS.
- υ OTRO ALGORITMO PARA MANEJAR EL CACHE DE ARCHIVOS DEL CLIENTE ES EL DE **ESCRITURA AL CIERRE**:
 - υ SE ADOPTA LA SEMANTICA DE SESION.
 - υ SOLO SE ESCRIBE UN ARCHIVO NUEVAMENTE EN EL SERVIDOR CUANDO EL ARCHIVO SE CIERRA:
 - ⊕ SE PODRIA ESPERAR (EJ: 30") PARA VER SI EL ARCHIVO ES ELIMINADO EN ESE LAPSO.
- υ UN METODO DISTINTO A LA CONSISTENCIA ES UTILIZAR UN ALGORITMO DE **CONTROL CENTRALIZADO**:
 - υ AL ABRIR UN ARCHIVO LA MAQUINA ENVIA UN MENSAJE AL SERVIDOR PARA ANUNCIAR ESTE HECHO.
 - υ EL SERVIDOR DE ARCHIVOS TIENE UN REGISTRO DE LOS ARCHIVOS ABIERTOS, SUS POSEEDORES Y SI ESTAN ABIERTOS PARA LECTURA, ESCRITURA O AMBOS PROCESOS.
 - υ SI SE ABRE UN ARCHIVO PARA LECTURA OTROS PROCESOS LO PUEDEN ABRIR PARA LECTURA PERO NO PARA ESCRITURA.
 - υ SI SE ABRE UN ARCHIVO PARA ESCRITURA SE DEBE EVITAR ABRIRLO PARA LECTURA DESDE OTRO PROCESO.
 - υ AL CERRAR UN ARCHIVO:
 - ⊕ SE DEBE INFORMAR AL SERVIDOR PARA QUE ACTUALICE SUS TABLAS.
 - ⊕ SE PUEDE ENVIAR EL ARCHIVO MODIFICADO AL SERVIDOR.

REPLICA

- v FRECUENTEMENTE LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS PROPORCIONAN LA REPLICA DE ARCHIVOS COMO UN SERVICIO:
 - u EXISTEN VARIAS COPIAS DE ALGUNOS ARCHIVOS.
 - u CADA COPIA ESTA EN UN SERVIDOR DE ARCHIVOS INDEPENDIENTE.
- v LAS PRINCIPALES RAZONES PARA LA REPLICA SON:
 - u AUMENTAR LA CONFIABILIDAD AL DISPONER DE RESPALDOS INDEPENDIENTES DE CADA ARCHIVO.
 - u PERMITIR EL ACCESO A ARCHIVOS AUN CUANDO FALLE UN SERVIDOR DE ARCHIVOS.
 - u REPARTIR LA CARGA DE TRABAJO ENTRE VARIOS SERVIDORES.
- v UN SISTEMA ES **TRANSPARENTE CON RESPECTO A LA REPLICA** SI LA MISMA SE ADMINISTRA SIN INTERVENCION DEL USUARIO.
- v UNA FORMA DE LLEVAR A CABO LA REPLICA CONSISTE EN QUE EL PROGRAMADOR CONTROLE TODO EL PROCESO (*REPLICA EXPLICITA*):
 - u LOS ARCHIVOS Y LAS COPIAS ADICIONALES SE CREAN EN SERVIDORES ESPECIFICOS.
 - u LAS DIRECCIONES EN LA RED DE TODAS LAS COPIAS SE ASOCIAN CON EL NOMBRE DEL ARCHIVO.
- v UN METODO ALTERNATIVO ES LA *REPLICA RETRASADA*:
 - u SOLO SE CREA UNA COPIA DE CADA ARCHIVO EN UN SERVIDOR.
 - u EL SERVIDOR CREA REPLICAS EN OTROS SERVIDORES:
 - Φ A POSTERIORI, AUTOMATICAMENTE Y SIN INTERVENCION DEL PROCESO DE USUARIO.
- v OTRO METODO CONSISTE EN EL USO DE LA *COMUNICACION EN GRUPO*:
 - u TODAS LAS OPERACIONES DE ESCRITURA SE TRANSMITEN SIMULTANEAMENTE A TODOS LOS SERVIDORES.
 - u LAS COPIAS ADICIONALES SE HACEN AL MISMO TIEMPO QUE EL ORIGINAL.

REPLICA

- v **PROTOCOLOS DE ACTUALIZACION.**
- v EL PRINCIPAL PROBLEMA ES ASEGURAR LA SINCRONIZACION DE LAS DISTINTAS COPIAS.
- v UN ALGORITMO POSIBLE ES EL DE **REPLICA DE LA COPIA PRIMARIA:**
 - u UNO DE LOS SERVIDORES SE DENOMINA COMO PRIMARIO.
 - u LOS DEMAS SERVIDORES SON SECUNDARIOS.
 - u LA ACTUALIZACION SE ENVIA AL SERVIDOR PRIMARIO:
 - Φ REALIZA LOS CAMBIOS LOCALMENTE.
 - Φ ENVIA COMANDOS A LOS SERVIDORES SECUNDARIOS PARA ORDENARLES LA MISMA MODIFICACION.
 - u LAS LECTURAS SE PUEDEN HACER DE CUALQUIER COPIA.
 - u LA DESVENTAJA ES QUE SI FALLA EL PRIMARIO NO SE PUEDEN LLEVAR A CABO LAS ACTUALIZACIONES.
- v OTRO POSIBLE ALGORITMO ES EL DEL **VOTO O DE GIFFORD:**
 - u LA IDEA FUNDAMENTAL ES EXIGIR A LOS CLIENTES QUE SOLICITEN Y ADQUIERAN EL PERMISO DE VARIOS SERVIDORES ANTES DE LEER O ESCRIBIR UN ARCHIVO REPLICADO.
 - u SE UTILIZA EL N° DE VERSION:
 - Φ IDENTIFICA LA VERSION DEL ARCHIVO Y ES LA MISMA PARA TODOS LOS ARCHIVOS RECIEN ACTUALIZADOS.
 - u PARA LEER UN ARCHIVO CON “N” REPLICAS UN CLIENTE DEBE CONFORMAR UN **QUORUM DE LECTURA:**
 - Φ COLECCION ARBITRARIA DE “N_r” SERVIDORES O MAS.
 - u PARA MODIFICAR UN ARCHIVO SE NECESITA UN **QUORUM DE ESCRITURA** DE AL MENOS “N_w” SERVIDORES.
 - u SE DEBE CUMPLIR QUE “N_r” + “N_w” > “N”:
 - Φ NUNCA SE PODRA OBTENER UN Q. DE LECTURA Y OTRO DE ESCRITURA AL MISMO TIEMPO.
 - u GENERALMENTE “N_r” ES MUY PEQUEÑO Y “N_w” MUY CERCANO A “N” YA QUE GENERALMENTE LAS LECTURAS SON MAS FRECUENTES QUE LAS ESCRITURAS.
 - u UNA VARIANTE ES EL ALGORITMO DEL **VOTO CON FANTASMA:**
 - Φ CREA UN SERVIDOR FANTASMA PARA CADA SERVIDOR REAL FALLIDO:
 - INTERVIENE SOLO EN EL Q. DE ESCRITURA.
 - LA ESCRITURA SOLO TIENE EXITO SI AL MENOS UNO DE LOS SERVIDORES ES REAL.

CONCLUSIONES IMPORTANTES RESPECTO DE LA IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS

- v LOS PRINCIPIOS GENERALMENTE CONSIDERADOS FUNDAMENTALES DEL DISEÑO DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS SON:
 - u LAS ESTACIONES DE TRABAJO TIENEN CICLOS QUE HAY QUE UTILIZAR:
 - ⊕ SI SE TIENE LA OPCION DE HACER ALGO EN UNA ESTACION DE TRABAJO O EN UN SERVIDOR:
 - ELEGIR LA ESTACION DE TRABAJO.
 - LOS CICLOS DE CPU DE LA ESTACION DE TRABAJO SON MENOS COSTOSOS QUE LOS CICLOS DE UN SERVIDOR.
 - u UTILIZAR EL CACHE EL MAXIMO POSIBLE:
 - ⊕ FRECUENTEMENTE AHORRAN CONSIDERABLE:
 - TIEMPO DE COMPUTO.
 - ANCHO DE BANDA DE LA RED.
 - u EXPLOTAR LAS PROPIEDADES DE USO:
 - ⊕ CONSIDERAR LA POSIBILIDAD DE IMPLANTAR TRATAMIENTOS DIFERENCIALES PARA LOS ARCHIVOS TRANSITORIOS DE CORTA VIDA Y NO COMPARTIDOS.
 - ⊕ TENER PRESENTE LA DIFICULTAD DE HABILITAR DIFERENTES VIAS PARA HACER LO MISMO.
 - ⊕ CONSIDERAR ASPECTOS TALES COMO EFICIENCIA Y SENCILLEZ.
 - u MINIMIZAR EL CONOCIMIENTO Y MODIFICACION A LO LARGO DEL SISTEMA:
 - ⊕ ES IMPORTANTE PARA LOGRAR ESCALABILIDAD.
 - ⊕ GENERALMENTE SON UTILES EN ESTE SENTIDO LOS DISEÑOS JERARQUICOS.
 - u CONFIAR EN EL MENOR N° POSIBLE DE ENTIDADES:
 - ⊕ SE TRATA DE UN PRINCIPIO YA ESTABLECIDO EN EL MUNDO DE LA SEGURIDAD.
 - u CREAR LOTES DE TRABAJO MIENTRAS SEA POSIBLE:
 - ⊕ EL USO DEL PROCESAMIENTO POR LOTES PUEDE CONTRIBUIR A UN MEJOR DESEMPEÑO.

TENDENCIAS EN LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS

CONSIDERACIONES RESPECTO DEL HARDWARE

- v **TENDENCIAS EN LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS.**
- v ES PROBABLE QUE LOS CAMBIOS EN EL HARDWARE TENGAN UN EFECTO MUY IMPORTANTE EN LOS FUTUROS SISTEMAS DISTRIBUIDOS DE ARCHIVOS.
- v TAMBIEN ES PROBABLE EL IMPACTO DEL CAMBIO EN LAS EXPECTATIVAS DEL USUARIO.
- v **CONSIDERACIONES RESPECTO DEL HARDWARE.**
- v EL ABARATAMIENTO DE LA MEMORIA PRINCIPAL PERMITIRA DISPONER DE SERVIDORES CON MEMORIAS CADA VEZ MAYORES:
 - u SE PODRIA ALOJAR DIRECTAMENTE EN MEMORIA EL SISTEMA DE ARCHIVOS LOGRANDO MAYOR SENCILLEZ Y DESEMPEÑO.
 - u SE DEBERIA PREVER LA OBTENCION DE RESPALDOS CONTINUOS O POR INCREMENTOS ANTE LA POSIBILIDAD DEL CORTE EN EL SUMINISTRO ELECTRICO.
 - u EL RESPALDO PODRIA HACERSE EN DISCOS OPTICOS REGRABABLES QUE TENGAN UNA ASOCIACION UNO A UNO CON LA MEMORIA:
 - Φ EL BYTE “k” DE LA MEMORIA CORRESPONDERIA AL BYTE “k” DEL DISCO.
- v LA DISPONIBILIDAD DE REDES DE FIBRA OPTICA DE ALTA VELOCIDAD PERMITIRIA ESQUEMAS TALES COMO:
 - u UN SERVIDOR DE ARCHIVOS EN LA MEMORIA PRINCIPAL DEL SERVIDOR CON RESPALDO EN EL DISCO OPTICO.
 - u ELIMINACION DEL DISCO DEL SERVIDOR Y DEL CACHE DEL CLIENTE.
 - u SE SIMPLIFICARIA SIGNIFICATIVAMENTE EL SOFTWARE.
- v LA POSIBLE CONSTRUCCION DE INTERFACES DE RED ESPECIALIZADAS QUE PERMITAN RESOLVER POR HARDWARE PROBLEMAS DIFICILES DE SOPORTAR POR SOFTWARE:
 - u CADA INTERFAZ DE RED TENDRIA UN MAPA DE BITS CON UN BIT POR CADA ARCHIVO EN EL CACHE.
 - u SE PODRIAN HABILITAR CERRADURAS POR ARCHIVO.
 - u PARA MODIFICAR UN ARCHIVO UN PROCESADOR ACTIVARIA EL BIT CORRESPONDIENTE EN LA INTERFAZ.

ESCALABILIDAD

REDES EN UN AREA AMPLIA

- v **ESCALABILIDAD.**
- v UNA TENDENCIA DEFINIDA EN LOS SISTEMAS DISTRIBUIDOS ES HACIA LOS SISTEMAS CADA VEZ MAS GRANDES.
- v LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS QUE OPERAN BIEN PARA CIENTOS DE MAQUINAS PODRIAN FALLAR EN ALGUN ASPECTO TRABAJANDO CON MILES O DECENAS DE MILES DE MAQUINAS.
- v GENERALMENTE LOS ALGORITMOS CENTRALIZADOS NO SE ESCALAN BIEN YA QUE EL SERVIDOR CENTRALIZADO PODRIA CONVERTIRSE EN UN CUELLO DE BOTELLA:
 - o SE PODRIA SEPARAR EL SISTEMA EN UNIDADES MAS PEQUEÑAS RELATIVAMENTE INDEPENDIENTES ENTRE SI.
- v LAS TRANSMISIONES TAMBIEN SON UN AREA PROBLEMÁTICA:
 - o SI CADA MAQUINA TRANSMITE UNA VEZ POR SEGUNDO:
 - φ CON “n” MAQUINAS HABRIA “n” TRANSMISIONES Y “n²” INTERRUPCIONES POR SEGUNDO.
 - φ SI “n” CRECE ESTO SE PUEDE CONVERTIR EN UN PROBLEMA.
- v EN GENERAL LOS RECURSOS Y ALGORITMOS NO DEBEN SER LINEALES CON RESPECTO AL N° DE USUARIOS.
- v **REDES EN UN AREA AMPLIA.**
- v GENERALMENTE LOS SISTEMAS DISTRIBUIDOS SE ASOCIAN CON REDES DE AREA LOCAL (LAN):
 - o CADA VEZ SERA MAYOR LA NECESIDAD DE CONECTARLOS ENTRE SI CUBRIENDO GRANDES AREAS (NACIONALES, REGIONALES, CONTINENTALES, ETC.).
- v LOS SISTEMAS DE ARCHIVOS DEBERAN SOPORTAR ESTAS NECESIDADES TENIENDO PRESENTE LA HETEROGENEIDAD DE LOS EQUIPOS, CODIGOS DE REPRESENTACION (ASCII, EBCDIC, ETC.), FORMATOS, ETC.
- v DEBERA ATENDERSE A LOS CAMBIOS DE TENDENCIA EN LOS REQUERIMIENTOS DE LAS APLICACIONES.
- v UN PROBLEMA ADICIONAL E INHERENTE EN LOS SISTEMAS DISTRIBUIDOS MASIVOS ES EL ANCHO DE BANDA DE LA RED:
 - o PUEDE RESULTAR INSUFICIENTE PARA EL DESEMPEÑO ESPERADO.

TOLERANCIA DE FALLOS

- v **USUARIOS MOVILES.**
- v LOS USUARIOS DE EQUIPOS MOVILES (LAPTOP, NOTEBOOK, ETC.) ESTAN GRAN PARTE DEL TIEMPO DESCONECTADOS DEL SISTEMA DE ARCHIVOS DE SU ORGANIZACION:
 - u REQUIEREN UNA SOLUCION, QUE PODRIA USAR OCULTAMIENTO:
 - ⊕ CUANDO ESTA CONECTADO EL USUARIO CARGA AL EQUIPO MOVIL LOS ARCHIVOS QUE CREE NECESITARA DESPUES.
 - ⊕ LOS UTILIZA MIENTRAS ESTA DESCONECTADO.
 - ⊕ AL RECONECTARSE, LOS ARCHIVOS EN EL CACHE DEBEN FUSIONARSE CON LOS EXISTENTES EN EL ARBOL DE DIRECTORIOS, LOGRANDO LA SINCRONIZACION.
 - ⊕ LA CONEXION PARA LA SINCRONIZACION PUEDE SER PROBLEMATICA SI SE UTILIZA UN ENLACE DE ANCHO DE BANDA REDUCIDO.
- v LO DESEABLE SERIA UN SISTEMA DISTRIBUIDO TOTALMENTE TRANSPARENTE PARA SU USO SIMULTANEO POR PARTE DE MILLONES DE USUARIOS MOVILES QUE FRECUENTEMENTE SE DESCONECTEN.
- v **TOLERANCIA DE FALLOS.**
- v LA DIFUSION DE LOS SISTEMAS DISTRIBUIDOS INCREMENTA LA DEMANDA DE SISTEMAS QUE ESENCIALMENTE NUNCA FALLEN.
- v LOS SISTEMAS TOLERANTES A FALLOS REQUERIRAN CADA VEZ MAS UNA CONSIDERABLE REDUNDANCIA EN:
 - u HARDWARE, COMUNICACIONES, SOFTWARE, DATOS, ETC.
- v LA REPLICA DE ARCHIVOS SERIA UN REQUISITO ESENCIAL.
- v TAMBIEN DEBERIA CONTEMPLARSE LA POSIBILIDAD DE QUE LOS SISTEMAS FUNCIONEN AUN CON LA CARENCIA DE PARTE DE LOS DATOS.
- v LOS TIEMPOS DE FALLO ACEPTABLES POR LOS USUARIOS SERAN CADA VEZ MENORES.

RENDIMIENTO

- v INTRODUCCION A LA MEDICION, CONTROL Y EVALUACION DEL RENDIMIENTO
- v TENDENCIAS IMPORTANTES QUE AFECTAN A LOS ASPECTOS DEL RENDIMIENTO
- v NECESIDAD DEL CONTROL Y DE LA EVALUACION DEL RENDIMIENTO
- v MEDICIONES DEL RENDIMIENTO
- v TECNICAS DE EVALUACION DEL RENDIMIENTO
- v EMBOTELLAMIENTOS Y SATURACION
- v CICLOS DE RETROALIMENTACION

INTRODUCCION A LA MEDICION, CONTROL Y EVALUACION DEL RENDIMIENTO

- v UN SISTEMA OPERATIVO ES EN PRIMER LUGAR UN ADMINISTRADOR DE RECURSOS:
 - u ES IMPORTANTE PODER DETERMINAR CON QUE EFECTIVIDAD ADMINISTRA SUS RECURSOS UN SISTEMA DETERMINADO.
- v GENERALMENTE HAY UN GRAN POTENCIAL DE MEJORA EN EL USO DE LOS RECURSOS EXISTENTES, PERO:
 - u MUCHAS INSTALACIONES REALIZAN MUY POCO O NINGUN CONTROL Y EVALUACION.
 - u CUANDO SE HACEN CONTROLES ESPECIFICOS SE GENERAN GRANDES CANTIDADES DE DATOS QUE MUCHAS VECES NO SE SABE COMO INTERPRETAR.
- v LAS INSTALACIONES RARA VEZ CUENTAN CON PERSONAL VERSADO EN LAS TECNICAS DE ANALISIS DE RENDIMIENTO.
- v DURANTE LOS PRIMEROS AÑOS DEL DESARROLLO DE LAS COMPUTADORAS:
 - u EL HARDWARE REPRESENTABA EL COSTO DOMINANTE DE LOS SISTEMAS:
 - Φ LOS ESTUDIOS DE RENDIMIENTO SE CONCENTRABAN EN EL HARDWARE.
- v ACTUALMENTE Y SEGUN LA TENDENCIA APRECIABLE:
 - u EL SOFTWARE REPRESENTA UNA PORCION CADA VEZ MAYOR DE LOS PRESUPUESTOS INFORMATICOS.
 - u EL SOFTWARE INCLUYE:
 - Φ S. O. DE MULTIPROGRAMACION / MULTIPROCESO, SISTEMAS DE COMUNICACIONES DE DATOS, SISTEMAS DE ADMINISTRACION DE BASES DE DATOS, SISTEMAS DE APOYO A VARIAS APLICACIONES, ETC.
 - u EL SOFTWARE FRECUENTEMENTE OCULTA EL HARDWARE AL USUARIO CREANDO UNA MAQUINA VIRTUAL:
 - Φ ESTA DEFINIDA POR LAS CARACTERISTICAS OPERATIVAS DEL SOFTWARE.
- v UN SOFTWARE DEFICIENTE Y / O MAL UTILIZADO PUEDE SER CAUSA DE UN RENDIMIENTO POBRE DEL HARDWARE:
 - u ES IMPORTANTE CONTROLAR Y EVALUAR EL RENDIMIENTO DEL HARDWARE Y DEL SOFTWARE.

TENDENCIAS IMPORTANTES QUE AFECTAN A LOS ASPECTOS DEL RENDIMIENTO

- v CON LOS AVANCES EN LA TECNOLOGIA DE HARDWARE LOS COSTOS DEL MISMO HAN DECRECIDO DRASTICAMENTE:
 - υ TODO HACE SUPONER QUE ESTA TENDENCIA CONTINUARA.
- v LOS COSTOS DE TRABAJO (PERSONAL) HAN IDO AUMENTANDO:
 - υ SIGNIFICAN UN PORCENTAJE IMPORTANTE DEL COSTO DE LOS SISTEMAS INFORMATICOS.
 - υ SE DEBE REFORMULAR EL ASPECTO DEL RENDIMIENTO DEL HARDWARE BASE Y MEDIRLO DE MANERA MAS ADAPTADA A LA PRODUCTIVIDAD HUMANA.
- v EL ADVENIMIENTO DEL MICROPROCESADOR EN LA DECADA DE 1.970:
 - υ HA PERMITIDO BAJAR CONSIDERABLEMENTE EL COSTO DE LOS CICLOS DE CPU.
 - υ HA DESPLAZADO EL FOCO DE ATENCION DE LA EVALUACION DEL RENDIMIENTO A OTRAS AREAS DONDE LOS COSTOS NO DISMINUYERON PROPORCIONALMENTE:
 - Φ EJ.: UTILIZACION DE DISPOSITIVOS DE ENTRADA / SALIDA.
- v TAMBIEN INFLUYEN EN LOS PUNTOS DE VISTA SOBRE LA EVALUACION DEL RENDIMIENTO ASPECTOS TALES COMO:
 - υ CONSTRUCCION DE REDES.
 - υ PROCESAMIENTO DISTRIBUIDO.
- v LAS CONEXIONES SE HACEN CON REDES Y NO SOLO CON COMPUTADORAS ESPECIFICAS:
 - υ SE PUEDE DISPONER DE CIENTOS O MILES DE SISTEMAS DE COMPUTACION.
 - υ SE PUEDE ACCEDER A COMPLEJOS SISTEMAS DE COMUNICACIONES DE DATOS.

NECESIDAD DEL CONTROL Y DE LA EVALUACION DEL RENDIMIENTO

- v LOS OBJETIVOS CORRIENTES EN LA EVALUACION DEL RENDIMIENTO GENERALMENTE SON:
 - u *EVALUACION DE SELECCION:*
 - Φ EL EVALUADOR DEBE DECIDIR SI LA ADQUISICION DE UN SISTEMA DE COMPUTACION ES APROPIADA.
 - u *PROYECCION DEL RENDIMIENTO:*
 - Φ EL EVALUADOR DEBE ESTIMAR EL RENDIMIENTO DE UN SISTEMA INEXISTENTE:
 - NUEVO SISTEMA O NUEVO COMPONENTE DE HARDWARE O DE SOFTWARE.
 - u *CONTROL DEL RENDIMIENTO:*
 - Φ EL EVALUADOR ACUMULA DATOS DEL RENDIMIENTO DE UN SISTEMA O COMPONENTE EXISTENTE PARA:
 - ASEGURAR QUE EL SISTEMA CUMPLE CON SUS METAS DE RENDIMIENTO.
 - AYUDAR A ESTIMAR EL IMPACTO DE LOS CAMBIOS PLANEADOS.
 - PROPORCIONAR LOS DATOS NECESARIOS PARA TOMAR DECISIONES ESTRATEGICAS.
- v EN LAS PRIMERAS FASES DEL DESARROLLO DE UN NUEVO SISTEMA SE INTENTA PREDECIR:
 - u LA NATURALEZA DE LAS APLICACIONES QUE CORRERAN EN EL SISTEMA.
 - u LAS CARGAS DE TRABAJO QUE LAS APLICACIONES DEBERAN MANEJAR.
- v DURANTE EL DESARROLLO E IMPLEMENTACION DE UN NUEVO SISTEMA SE INTENTA DETERMINAR:
 - u LA MEJOR ORGANIZACION DEL HARDWARE.
 - u LAS ESTRATEGIAS DE ADMINISTRACION DE RECURSOS QUE DEBERAN IMPLANTARSE EN EL S. O.
 - u SI EL SISTEMA CUMPLE O NO CON SUS OBJETIVOS DE RENDIMIENTO.
- v FRECUENTEMENTE SON NECESARIOS *PROCESOS DE CONFIGURACION* DE LOS SISTEMAS PARA QUE PUEDAN SERVIR A LAS NECESIDADES.
- v LOS *PROCESOS DE SINTONIZACION DEL SISTEMA* TIENDEN A MEJORAR EL RENDIMIENTO EN BASE A AJUSTAR EL SISTEMA A LAS CARACTERISTICAS DE LA INSTALACION DEL USUARIO.

MEDICIONES DEL RENDIMIENTO

- v EL RENDIMIENTO EXPRESA LA MANERA O LA EFICIENCIA CON QUE UN SISTEMA DE COMPUTACION CUMPLE SUS METAS.
- v EL RENDIMIENTO ES UNA *CANTIDAD RELATIVA* MAS QUE ABSOLUTA PERO SUELE HABLARSE DE *MEDIDAS ABSOLUTAS DE RENDIMIENTO*:
 - v EJ.: N° DE TRABAJOS ATENDIDOS POR UNIDAD DE TIEMPO.
- v ALGUNAS MEDICIONES SON DIFICILES DE CUANTIFICAR:
 - v EJ.: *FACILIDAD DE USO*.
- v OTRAS MEDICIONES SON FACILES DE CUANTIFICAR:
 - v EJ.: *ACCESOS A UN DISCO EN LA UNIDAD DE TIEMPO*.
- v LAS MEDICIONES DE RENDIMIENTO PUEDEN ESTAR:
 - v ORIENTADAS HACIA EL USUARIO:
 - Φ EJ.: TIEMPOS DE RESPUESTA.
 - v ORIENTADAS HACIA EL SISTEMA:
 - Φ EJ.: UTILIZACION DE LA CPU.
- v ALGUNAS MEDICIONES DEL RENDIMIENTO COMUNES SON:
 - v *TIEMPO DE REGRESO*:
 - Φ TIEMPO DESDE LA ENTREGA DEL TRABAJO HASTA SU REGRESO AL USUARIO (PARA PROCESAMIENTO POR LOTES).
 - v *TIEMPO DE RESPUESTA*:
 - Φ TIEMPO DE REGRESO DE UN SISTEMA INTERACTIVO.
 - v *TIEMPO DE REACCION DEL SISTEMA*:
 - Φ TIEMPO DESDE QUE EL USUARIO PRESIONA “ENTER” HASTA QUE SE DA LA PRIMERA SECCION DE TIEMPO DE SERVICIO.
- v LAS ANTERIORES SON CANTIDADES PROBABILISTICAS Y SE CONSIDERAN COMO VARIABLES ALEATORIAS EN LOS ESTUDIOS DE:
 - v SIMULACION.
 - v MODELADO DE SISTEMAS.

MEDICIONES DEL RENDIMIENTO

v OTRAS MEDIDAS DEL RENDIMIENTO UTILIZADAS SON:

u *VARIANZA DE LOS TIEMPOS DE RESPUESTA (O DE OTRA DE LAS VARIABLES ALEATORIAS CONSIDERADAS):*

- ⊕ ES UNA MEDIDA DE DISPERSION.
- ⊕ SI ES PEQUEÑA INDICA TIEMPOS PROXIMOS A LA MEDIA.
- ⊕ SI ES GRANDE INDICA TIEMPOS ALEJADOS DE LA MEDIA.
- ⊕ ES UNA MEDIDA DE LA PREDECIBILIDAD.

u *CAPACIDAD DE EJECUCION:*

- ⊕ ES LA MEDIDA DE LA EJECUCION DE TRABAJO POR UNIDAD DE TIEMPO.

u *CARGA DE TRABAJO:*

- ⊕ ES LA MEDIDA DE LA CANTIDAD DE TRABAJO QUE:
 - HA SIDO INTRODUCIDA EN EL SISTEMA.
 - EL SISTEMA DEBE PROCESAR NORMALMENTE PARA FUNCIONAR DE MANERA ACEPTABLE.

u *CAPACIDAD:*

- ⊕ ES LA MEDIDA DE LA CAPACIDAD DE RENDIMIENTO MAXIMA QUE UN SISTEMA PUEDE TENER SIEMPRE QUE:
 - EL SISTEMA ESTE LISTO PARA ACEPTAR MAS TRABAJOS.
 - HAYA ALGUNO INMEDIATAMENTE DISPONIBLE.

u *UTILIZACION:*

- ⊕ ES LA FRACCION DE TIEMPO QUE UN RECURSO ESTA EN USO.
- ⊕ ES DESEABLE UN GRAN PORCENTAJE DE UTILIZACION PERO ESTE PUEDE SER EL RESULTADO DE UN USO INEFICIENTE.
- ⊕ CUANDO SE APLICA A LA CPU SE DEBE DISTINGUIR ENTRE:
 - USO EN TRABAJOS PRODUCTIVOS DE APLICACIÓN.
 - USO EN SOBRECARGA DEL SISTEMA.

TECNICAS DE EVALUACION DEL RENDIMIENTO

v **TIEMPOS.**

- v LOS TIEMPOS PROPORCIONAN LOS MEDIOS PARA REALIZAR COMPARACIONES RAPIDAS DEL HARDWARE.
- v UNA POSIBLE UNIDAD DE MEDIDA ES EL “MIPS”:
 - u MILLON DE INSTRUCCIONES POR SEGUNDO.
- v LOS TIEMPOS SE USAN PARA COMPARACIONES RAPIDAS:
 - u SE UTILIZAN OPERACIONES BASICAS DE HARDWARE.

v **MEZCLAS DE INSTRUCCIONES.**

- v SE USA UN PROMEDIO PONDERADO DE VARIOS TIEMPOS DE LAS INSTRUCCIONES MAS APROPIADAS PARA UNA APLICACION DETERMINADA:
 - u LOS EQUIPOS PUEDEN SER COMPARADOS CON MAYOR CERTEZA DE LA QUE PROPORCIONAN LOS TIEMPOS POR SI SOLOS.
- v SON UTILES PARA COMPARACIONES RAPIDAS DEL HARDWARE.
- v **PROGRAMAS DEL NUCLEO.**
- v UN *PROGRAMA NUCLEO* ES UN PROGRAMA TIPICO QUE PUEDE SER EJECUTADO EN UNA INSTALACION.
- v SE UTILIZAN LOS TIEMPOS ESTIMADOS QUE SUMINISTRAN LOS FABRICANTES PARA CADA MAQUINA PARA CALCULAR SU TIEMPO DE EJECUCION.
- v SE CORRE EL PROGRAMA TIPICO EN LAS DISTINTAS MAQUINAS PARA OBTENER SU TIEMPO DE EJECUCION.
- v PUEDEN SER UTILES PARA LA EVALUACION DE CIERTOS COMPONENTES DEL SOFTWARE, POR EJ. COMPILADORES:

- u PUEDEN AYUDAR A DETERMINAR QUE COMPILADOR GENERA EL CODIGO MAS EFICIENTE.

v **MODELOS ANALITICOS.**

- v SON REPRESENTACIONES MATEMATICAS DE SISTEMAS DE COMPUTACION O DE COMPONENTES DE SISTEMAS DE COMPUTACION.
- v GENERALMENTE SE UTILIZAN LOS MODELOS DE:
 - u *TEORIA DE COLAS.*
 - u *PROCESOS DE MARKOV.*
- v REQUIEREN UN GRAN NIVEL MATEMATICO DEL EVALUADOR Y SON CONFIABLES SOLO EN SISTEMAS SENCILLOS:
 - u EN SISTEMAS COMPLEJOS LOS SUPUESTOS SIMPLIFICADORES PUEDEN INVALIDAR SU UTILIDAD Y APLICABILIDAD.

TECNICAS DE EVALUACION DEL RENDIMIENTO

- v **PUNTOS DE REFERENCIA (O PROGRAMAS DE COMPARACION DEL RENDIMIENTO).**
- v SON PROGRAMAS REALES QUE EL EVALUADOR EJECUTA EN LA MAQUINA QUE SE ESTA EVALUANDO.
- v GENERALMENTE ES UN *PROGRAMA DE PRODUCCION*:
 - u TIPICO DE MUCHOS TRABAJOS DE LA INSTALACION.
 - u QUE SE EJECUTA CON REGULARIDAD.
- v EL PROGRAMA COMPLETO SE EJECUTA EN LA MAQUINA REAL CON DATOS REALES.
- v SE DEBEN SELECCIONAR CUIDADOSAMENTE LOS *PUNTOS DE REFERENCIA* PARA QUE SEAN REPRESENTATIVOS DE LOS TRABAJOS DE LA INSTALACION.
- v LOS EFECTOS DEL SOFTWARE PUEDEN EXPERIMENTARSE DIRECTAMENTE EN VEZ DE ESTIMARSE.
- v **PROGRAMAS SINTETICOS.**
- v COMBINAN LAS TECNICAS DE LOS NUCLEOS Y LOS PUNTOS DE REFERENCIA.
- v SON PROGRAMAS REALES DISEÑADOS PARA EJERCITAR CARACTERISTICAS ESPECIFICAS DE UNA MAQUINA.
- v **SIMULACION.**
- v ES UNA TECNICA CON LA CUAL EL EVALUADOR DESARROLLA UN MODELO COMPUTARIZADO DEL SISTEMA QUE SE ESTA EVALUANDO.
- v ES POSIBLE PREPARAR UN MODELO DE UN SISTEMA INEXISTENTE Y EJECUTARLO PARA VER COMO SE COMPORTARIA EN CIERTAS CIRCUNSTANCIAS:
 - u SE PUEDE EVITAR LA CONSTRUCCION DE SISTEMAS MAL DISEÑADOS.
- v LOS SIMULADORES SON MUY APLICADOS EN LAS INDUSTRIAS ESPACIAL Y DE TRANSPORTES.
- v LOS SIMULADORES PUEDEN SER:
 - u *MANEJADOS POR EVENTOS*:
 - Φ SON CONTROLADOS POR LOS EVENTOS PRODUCIDOS EN EL SIMULADOR SEGUN DISTRIBUCIONES PROBABILISTICAS.
 - u *MANEJADOS POR LIBRETO*:
 - Φ SON CONTROLADOS POR DATOS OBTENIDOS DE FORMA EMPIRICA Y MANIPULADOS CUIDADOSAMENTE PARA REFLEJAR EL COMPORTAMIENTO ANTICIPADO DEL SISTEMA SIMULADO.

TECNICAS DE EVALUACION DEL RENDIMIENTO

- v **CONTROL DEL RENDIMIENTO.**
- v ES LA RECOLECCION Y ANALISIS DE INFORMACION RELATIVA AL RENDIMIENTO DEL SISTEMA EXISTENTE.
- v PERMITE LOCALIZAR EMBOTELLAMIENTOS CON RAPIDEZ.
- v PUEDE AYUDAR A DECIDIR LA FORMA DE MEJORAR EL RENDIMIENTO.
- v PUEDE SER UTIL PARA DETERMINAR LA DISTRIBUCION DE TRABAJOS DE VARIOS TIPOS:
 - v PERMITIRIA ACONSEJAR EL USO DE *COMPILADORES OPTIMIZADORES* O *COMPILADORES RAPIDOS Y SUCIOS*.
- v EL CONTROL DEL RENDIMIENTO PUEDE HACERSE POR MEDIO DE TECNICAS DE HARDWARE O DE SOFTWARE.
- v **LOS MONITORES DE SOFTWARE:**
 - v GENERALMENTE SON ECONOMICOS.
 - v PUEDEN DISTORSIONAR LAS LECTURAS DEL RENDIMIENTO DEBIDO A QUE CONSUMEN RECURSOS DEL SISTEMA.
- v **LOS MONITORES DE HARDWARE:**
 - v GENERALMENTE SON MAS COSTOSOS.
 - v SU INFLUENCIA SOBRE LA OPERACION DEL SISTEMA ES MINIMA.
- v **LOS MONITORES:**
 - v PRODUCEN GRANDES CANTIDADES DE DATOS QUE DEBEN SER ANALIZADOS MANUALMENTE O POR SISTEMA.
 - v INDICAN CON PRECISION COMO ESTA FUNCIONANDO UN SISTEMA.
 - v SON DE MUCHA AYUDA PARA EVALUAR SISTEMAS EN DESARROLLO Y TOMAR LAS DECISIONES DE DISEÑO ADECUADAS.
- v **LOS RASTROS DE EJECUCION DE INSTRUCCIONES (TRACE) O RASTROS DE EJECUCION DE MODULOS** PUEDEN REVELAR EMBOTELLAMIENTOS.
- v UN RASTREO DE EJECUCION DE MODULOS PUEDE MOSTRAR QUE SE ESTA EJECUTANDO UN PEQUEÑO SUBCONJUNTO DE MODULOS DURANTE GRAN PARTE DEL TIEMPO:
 - v LOS DISEÑADORES DEBERAN OPTIMIZARLOS PARA MEJORAR EN GRAN MEDIDA EL RENDIMIENTO DEL SISTEMA.
 - v SE PODRIA ELIMINAR EL COSTO DE OPTIMIZACION DE LOS MODULOS POCO USADOS.

EMBOTELLAMIENTOS

SATURACION

Y

- v LOS RECURSOS ADMINISTRADOS POR LOS S. O. SE ACOPLAN E INTERACTUAN DE MANERAS COMPLEJAS PARA AFECTAR AL TOTAL DE LA OPERACION DEL SISTEMA.
- v CIERTOS RECURSOS PUEDEN SUFRIR EMBOTELLAMIENTOS QUE LIMITAN EL RENDIMIENTO DEL SISTEMA:
 - v NO PUEDEN REALIZAR SU PARTE DEL TRABAJO.
 - v OTROS RECURSOS PUEDEN ESTAR CON EXCESO DE CAPACIDAD.
- v UN EMBOTELLAMIENTO TIENDE A PRODUCIRSE EN UN RECURSO CUANDO EL TRAFICO DE TRABAJOS O PROCESOS DE ESE RECURSO COMIENZA A *ALCANZAR SU CAPACIDAD LIMITE*:
 - v EL RECURSO SE ENCUENTRA *SATURADO*.
 - v LOS PROCESOS QUE COMPITEN POR EL RECURSO COMIENZAN A INTERFERIRSE UNOS A OTROS.
 - v EJ.: PROBLEMA DE LA HIPERPAGINACION:
 - Φ OCURRE CUANDO EL ALMACENAMIENTO PRINCIPAL ESTA LLENO.
 - Φ LOS CONJUNTOS DE TRABAJO DE LOS DISTINTOS PROCESOS ACTIVOS NO PUEDEN SER MANTENIDOS SIMULTANEAMENTE EN EL ALMACENAMIENTO PRINCIPAL.
- v PARA DETECTAR LOS EMBOTELLAMIENTOS SE DEBE CONTROLAR CADA COLA DE PETICIONES DE LOS RECURSOS:
 - v CUANDO UNA COLA CRECE RAPIDAMENTE SIGNIFICA QUE LA *TASA DE LLEGADAS DE PETICIONES* DEBE SUPERAR A SU *TASA DE SERVICIO*.
- v EL AISLAMIENTO DE LOS EMBOTELLAMIENTOS ES UNA PARTE IMPORTANTE DE LA AFINACION DE LA SINTONIZACION DEL SISTEMA.
- v LOS EMBOTELLAMIENTOS PUEDEN ELIMINARSE:
 - v AUMENTANDO LA CAPACIDAD DE LOS RECURSOS.
 - v AÑADIENDO MAS RECURSOS DE ESE TIPO EN ESE PUNTO DEL SISTEMA.

CICLOS DE RETROALIMENTACION

- v EL RENDIMIENTO DE UN S. O. PUEDE SER SENSIBLE AL ESTADO ACTUAL DEL SISTEMA.
- v UN *CICLO DE RETROALIMENTACION* ES UNA SITUACION EN LA CUAL LA INFORMACION DEL ESTADO ACTUAL DEL SISTEMA SE PONE A DISPOSICION DE LAS PETICIONES ENTRANTES.
- v LA RUTA DE LAS PETICIONES PUEDE MODIFICARSE:
 - u SI LA RETROALIMENTACION INDICA QUE PUEDE HABER DIFICULTAD DE DARLES SERVICIO.
- v **RETROALIMENTACION NEGATIVA.**
- v LA TASA DE LLEGADAS DE NUEVAS PETICIONES PUEDE DECRECER COMO RESULTADO DE LA INFORMACION QUE SE ESTA RETROALIMENTANDO.
- v CONTRIBUYE A LA *ESTABILIDAD* DE LOS SISTEMAS DE COLAS:
 - u IMPIDE QUE LAS COLAS CREZCAN INDEFINIDAMENTE.
 - u HACE QUE LA LONGITUD DE LAS COLAS SE MANTENGA CERCA DE SUS VALORES MEDIOS.
- v **RETROALIMENTACION POSITIVA.**
- v LA INFORMACION RETROALIMENTADA PROVOCA UN INCREMENTO EN VEZ DE UN DECREMENTO DE ALGUN PARAMETRO.
- v SE DEBEN EVITAR SITUACIONES SIMILARES A LA SIGUIENTE:
 - u EL S. O. DETECTA CAPACIDAD DISPONIBLE DE CPU.
 - u EL S. O. INFORMA AL PLANIFICADOR DE TRABAJOS QUE ADMITA MAS TRABAJOS EN LA MEZCLA DE MULTIPROGRAMACION:
 - Φ CON ESTO SE INCREMENTARIA EL USO DE CPU.
 - u AL INCREMENTARSE LA MEZCLA DE MULTIPROGRAMACION:
 - Φ DECRECE LA CANTIDAD DE MEMORIA QUE SE PUEDE ASIGNAR A CADA TRABAJO.
 - Φ EL N° DE FALLOS DE PAGINA PUEDE INCREMENTARSE.
 - Φ LA UTILIZACION DE CPU PUEDE DECRECER.
- v PUEDE PRODUCIR *INESTABILIDADES*:
 - u DEBE DISEÑARSE CON MUCHA PRUDENCIA.
 - u SE DEBEN CONTROLAR LOS EFECTOS DE CADA CAMBIO INCREMENTAL PARA VER SI RESULTA UNA MEJORA ANTICIPADA.
 - u SI UN CAMBIO INCREMENTAL DETERIORA EL RENDIMIENTO SE PODRIA ESTAR OPERANDO EN UN RANGO INESTABLE.

MODELADO ANALITICO EN RELACION AL RENDIMIENTO

- v INTRODUCCION AL MODELADO ANALITICO Y TEORIA DE COLAS
- v FUENTE, LLEGADAS Y LLEGADAS DE POISSON
- v TIEMPOS DE SERVICIO, CAPACIDAD DE LA COLA Y NUMERO DE SERVIDORES EN EL SISTEMA
- v DISCIPLINAS DE COLAS
- v INTENSIDAD DE TRAFICO Y UTILIZACION DEL SERVIDOR
- v ESTADO ESTABLE EN FUNCION DE SOLUCIONES TRANSITORIAS
- v RESULTADO DE LITTLE
- v RESUMEN DEL PROCESO DE POISSON
- v ANALISIS DE UN SISTEMA DE COLAS $M/M/1$
- v ANALISIS DE UN SISTEMA DE COLAS $M/M/C$
- v PROCESOS DE MARKOV
- v PROCESOS DE NACIMIENTO Y MUERTE
- v ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO

INTRODUCCION AL MODELADO ANALITICO Y TEORIA DE COLAS

- v ALGUNAS DE LAS TECNICAS MAS CONOCIDAS DE MODELADO ANALITICO SON:
 - u LA TEORIA DE COLAS.
 - u LOS PROCESOS DE MARKOV.
- v LOS MODELOS ANALITICOS:
 - u SON LAS REPRESENTACIONES MATEMATICAS DE LOS SISTEMAS.
 - u PERMITEN AL EVALUADOR DEL RENDIMIENTO SACAR CONCLUSIONES ACERCA DEL COMPORTAMIENTO DEL SISTEMA.
- v LAS EXPRESIONES *TEORIA DE COLAS Y TEORIA DE LINEAS (O FILAS) DE ESPERA* DEBEN CONSIDERARSE EQUIVALENTES.
- v EL TERMINO MATEMATICO *COLA* SIGNIFICA UNA *LINEA DE ESPERA*.
- v SI NO HUBIERA LINEAS DE ESPERA SE PODRIA RECIBIR SERVICIO DE INMEDIATO:
 - u SERIA LO DESEABLE.
 - u EL COSTO DE DISPONER DE LA SUFICIENTE CAPACIDAD DE SERVICIO PARA NO TENER QUE ESPERAR SERIA MUY ELEVADO.
- v SE CONSUME CIERTA CANTIDAD DE TIEMPO EN LINEAS DE ESPERA POR SERVICIO PERO:
 - u EL COSTO DE ESE SERVICIO ES MENOR DEBIDO A LA MEJOR UTILIZACION DE LA INSTALACION DE SERVICIO.
- v SI EXISTE UNA POBLACION DE CLIENTES QUE DEMANDAN CIERTO SERVICIO PRESTADO POR SERVIDORES:
 - u ALGUNOS CLIENTES INGRESARAN A LA RED DE COLAS Y ESPERARAN QUE UN SERVIDOR QUEDE DISPONIBLE.
- v ALGUNAS COLAS SON:
 - u *ILIMITADAS*: PUEDEN CRECER TANTO COMO SEA NECESARIO PARA CONTENER A LOS CLIENTES QUE ESPERAN.
 - u *LIMITADAS*: SOLO PUEDEN CONTENER UN N° FIJO DE CLIENTES EN ESPERA Y QUIZAS HASTA NINGUNO.
- v SE DEBEN TENER EN CUENTA *VARIABLES ALEATORIAS* QUE PUEDEN SER DESCRITAS POR *DISTRIBUCIONES PROBABILISTICAS*.
- v LA VARIABLE ALEATORIA “q” REPRESENTA EL TIEMPO QUE EMPLEA UN CLIENTE ESPERANDO EN LA COLA A SER SERVIDO.
- v LA VARIABLE ALEATORIA “s” REPRESENTA LA CANTIDAD DE TIEMPO QUE EMPLEA UN CLIENTE EN SER SERVIDO.
- v LA VARIABLE ALEATORIA “w” REPRESENTA EL TIEMPO TOTAL QUE EMPLEA UN CLIENTE EN EL SISTEMA DE COLAS: “ $w = q + s$ ”.

FUENTE, LLEGADAS Y LLEGADAS DE POISSON

✓ **FUENTE.**

- ✓ LOS CLIENTES SON PROPORCIONADOS A UN SISTEMA DE COLAS DESDE UNA *FUENTE* QUE PUEDE SER INFINITA O FINITA.
- ✓ CON UNA *FUENTE INFINITA* LA COLA DE SERVICIO PUEDE LLEGAR A SER ARBITRARIAMENTE GRANDE.
- ✓ PARA UNA *FUENTE FINITA* LA COLA DE SERVICIO ES LIMITADA.
 - ✓ UNA FUENTE FINITA PERO MUY GRANDE SUELE CONSIDERARSE COMO INFINITA.

✓ **LLEGADAS.**

- ✓ SUPONDREMOS QUE LOS CLIENTES LLEGAN A UN SISTEMA DE COLAS EN LOS TIEMPOS:
 - ✓ $t_0 < t_1 < t_2 < \dots < t_n$.
- ✓ LOS CLIENTES LLEGAN DE UNO EN UNO Y NUNCA HAY UNA COLISION.
- ✓ LAS VARIABLES ALEATORIAS “ τ_k ” MIDEN LOS TIEMPOS ENTRE LAS LLEGADAS SUCESIVAS (ARBITRARIO) Y SE DENOMINAN *TIEMPOS ENTRE LLEGADAS*:
 - ✓ SON VARIABLES ALEATORIAS *INDEPENDIENTES* Y ESTAN *IDENTICAMENTE DISTRIBUIDAS*.
 - ✓ $\tau_k = t_k - t_{k-1}$ ($k \geq 1$).

✓ **LLEGADAS DE POISSON.**

- ✓ LAS LLEGADAS PUEDEN SEGUIR DISTINTOS PATRONES ARBITRARIOS PERO SUELE SUPONERSE QUE FORMAN UN *PROCESO DE LLEGADAS DE POISSON*:
 - ✓ LOS TIEMPOS ENTRE LLEGADAS ESTAN DISTRIBUIDOS EXPONENCIALMENTE:
 - ✓ $P(\tau \leq t) = 1 - e^{-\lambda t}$
 - ✓ LA PROBABILIDAD DE QUE LLEGUEN EXACTAMENTE n CLIENTES EN CUALQUIER INTERVALO DE LONGITUD t ES:
 - ✓ $[e^{-\lambda t}(\lambda t)^n] / n!$ ($n = 0, 1, 2, \dots$).
 - ✓ λ ES UNA TASA PROMEDIO DE LLEGADAS CONSTANTE EXPRESADA EN “CLIENTES POR UNIDAD DE TIEMPO”.
 - ✓ EL N° DE LLEGADAS POR UNIDAD DE TIEMPO SE DICE QUE TIENE *DISTRIBUCION DE POISSON* CON UNA MEDIA λ .

TIEMPOS DE SERVICIO, CAPACIDAD DE LA COLA Y NUMERO DE SERVIDORES EN EL SISTEMA

- v **TIEMPOS DE SERVICIO.**
- v SE SUPONE QUE LOS TIEMPOS DE SERVICIO SON ALEATORIOS.
- v “ s_k ” ES EL TIEMPO DE SERVICIO QUE EL k -ESIMO CLIENTE REQUIERE DEL SISTEMA.
- v UN TIEMPO DE SERVICIO ARBITRARIO SE DESIGNA POR “ s ”.
- v LA DISTRIBUCION DE LOS TIEMPOS DE SERVICIO ES:
 - v $W_s(t) = P(s \leq t)$.
- v PARA UN SERVICIO ALEATORIO CON UNA TASA PROMEDIO DE SERVICIO “ μ ”:
 - v $W_s(t) = P(s \leq t) = 1 - e^{-\mu t} \quad (t \geq 0)$.
- v **CAPACIDAD DE LA COLA.**
- v LAS COLAS DEBEN TENER:
 - v *CAPACIDAD INFINITA:*
 - Φ CADA CLIENTE QUE LLEGUE PUEDE ENTRAR EN EL SISTEMA DE COLAS Y ESPERAR, INDEPENDIENTEMENTE DE CUANTOS CLIENTES HAY EN ESPERA.
 - v *CAPACIDAD CERO (O SISTEMAS DE PERDIDAS):*
 - Φ LOS CLIENTES QUE LLEGAN CUANDO LA INSTALACION DE SERVICIO ESTA OCUPADA NO PODRAN SER ADMITIDOS EL SISTEMA.
 - v *CAPACIDAD POSITIVA:*
 - Φ LOS CLIENTES QUE LLEGAN SOLO ESPERAN SI HAY LUGAR EN LA COLA.
- v **NUMERO DE SERVIDORES EN EL SISTEMA.**
- v LOS SISTEMAS DE COLAS SE PUEDEN CATEGORIZAR SEGUN EL N° DE SERVIDORES EN:
 - v *SISTEMAS DE UN SOLO SERVIDOR:*
 - Φ TIENEN UN SOLO SERVIDOR Y NADA MAS PUEDEN DARLE SERVICIO A UN SOLO CLIENTE A LA VEZ.
 - v *SISTEMAS DE SERVIDORES MULTIPLES:*
 - Φ TIENEN “ c ” SERVIDORES CON IDENTICA CAPACIDAD Y PUEDEN DAR SERVICIO A “ c ” CLIENTES A LA VEZ.

DISCIPLINAS DE COLAS

- v SON LAS REGLAS USADAS PARA ELEGIR AL SIGUIENTE CLIENTE DE COLA QUE VA A SER SERVIDO.
- v LA MAS CONOCIDA ES LA “FCFS” O PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO.
- v GENERALMENTE SE UTILIZAN LAS SIGUIENTES NOTACIONES:
 - u NOTACION KENDALL.
 - u NOTACION KENDALL ABREVIADA.
- v NOTACION KENDALL (A/B/c/K/m/Z):
 - u A: DISTRIBUCION DE TIEMPOS ENTRE LLEGADAS.
 - u B: DISTRIBUCION DE TIEMPOS DE SERVICIO.
 - u c: N° DE SERVIDORES.
 - u K: CAPACIDAD DE COLA DEL SISTEMA.
 - u m: N° DE CLIENTES EN LA FUENTE.
 - u Z: DISCIPLINA DE COLA.
- v NOTACION KENDALL ABREVIADA (A/B/c):
 - u NO HAY LIMITE EN LA LONGITUD DE LA COLA.
 - u LA FUENTE ES INFINITA.
 - u LA DISCIPLINA DE COLA ES “FCFS”.
 - u “A” Y “B” PUEDEN SER:
 - Φ GI: PARA TIEMPO ENTRE LLEGADAS GENERAL INDEPENDIENTE.
 - Φ G: PARA TIEMPO DE SERVICIO GENERAL.
 - Φ E_k : PARA LAS DISTRIBUCIONES DE TIEMPOS ENTRE LLEGADAS O DE SERVICIO ERLANG-k.
 - Φ M: PARA LAS DISTRIBUCIONES DE TIEMPOS ENTRE LLEGADAS O DE SERVICIO EXPONENCIALES.
 - Φ D: PARA LAS DISTRIBUCIONES DE TIEMPOS ENTRE LLEGADAS O DE SERVICIO DETERMINISTICOS.
 - Φ H_k : PARA LAS DISTRIBUCIONES DE TIEMPOS ENTRE LLEGADAS O DE SERVICIO HIPEREXPONENCIALES (CON “k” ESTADOS).

INTENSIDAD DE TRAFICO Y UTILIZACION DEL SERVIDOR

v **INTENSIDAD DE TRAFICO.**

v ES UNA MEDIDA DE LA CAPACIDAD DEL SISTEMA PARA DAR SERVICIO EFECTIVO A SUS CLIENTES.

v SE DEFINE COMO LA RAZON DE LA MEDIA DEL TIEMPO DE SERVICIO “E(s)” Y LA MEDIA DEL TIEMPO ENTRE LLEGADAS “E(τ)”.

v LA INTENSIDAD DE TRAFICO “u” ES:

u = [E(s)] / [E(τ)] = λ E(s) = (λ / μ):

Φ λ: TASA DE LLEGADAS.

Φ μ: TASA DE SERVICIO.

v ES UTIL PARA DETERMINAR EL N° MINIMO DE SERVIDORES IDENTICOS QUE NECESITARA UN SISTEMA PARA DAR SERVICIO A SUS CLIENTES:

u SIN QUE LAS COLAS SE HAGAN INDEFINIDAMENTE LARGAS.

u SIN TENER QUE RECHAZAR CLIENTES.

u EJ: SI E(s) = 17 SEGUNDOS Y E(τ) = 5 SEGUNDOS, u = 17 / 5 = 3,4:

Φ EL SISTEMA DEBERA TENER UN MINIMO DE 4 SERVIDORES.

v SE DEBE TENER EN CUENTA QUE:

u LA TASA DE LLEGADAS DE LOS CLIENTES ES UN PROMEDIO:

Φ ES POSIBLE QUE NO LLEGUE NINGUN CLIENTE DURANTE UN LARGO TIEMPO.

Φ ES POSIBLE QUE LOS CLIENTES LLEGUEN EN RAPIDA SUCESION, EXCEDIENDO LA CAPACIDAD FISICA DE LA COLA Y OCASIONANDO EL RECHAZO DE CLIENTES.

Φ SI SE UTILIZAN COLAS DE TAMAÑO FIJO DEBE HABER CAPACIDAD SUFICIENTE PARA SOPORTAR EXCESOS OCASIONALES DE LA TASA DE LLEGADAS.

Φ SE PODRIAN UTILIZAR COLAS DE LONGITUD VARIABLE JUNTO CON LISTA ENCADENADA.

v **UTILIZACION DEL SERVIDOR.**

v SE DEFINE COMO LA INTENSIDAD DE TRAFICO POR SERVIDOR:

ρ = u / c = λ / (μ c).

v ES LA PROBABILIDAD DE QUE UN SERVIDOR DETERMINADO SE ENCUENTRE OCUPADO.

v SEGUN LA *LEY DE LOS GRANDES NUMEROS* ESTA PROBABILIDAD ES APROXIMADAMENTE LA FRACCION DE TIEMPO QUE CADA SERVIDOR ESTA EN USO.

v EN SISTEMAS DE UN SOLO SERVIDOR ES IGUAL A LA INTENSIDAD DE TRAFICO.

ESTADO ESTABLE EN FUNCION DE SOLUCIONES TRANSITORIAS

RESULTADO DE LITTLE

- v **ESTADO ESTABLE EN FUNCION DE SOLUCIONES TRANSITORIAS.**
- v LOS SISTEMAS DE COLAS QUE SE HAN “ASENTADO” SE DICE QUE ESTAN OPERANDO EN *ESTADO ESTABLE*.
- v GENERALMENTE LA OPERACION INICIAL DE UN SISTEMA NO ES INDICATIVA DE SU COMPORTAMIENTO PERIODICO.
- v LOS SISTEMAS DE COLAS DEBEN PASAR POR ALGUN PERIODO INICIAL DE OPERACION ANTES DE TENER UN FUNCIONAMIENTO:
 - v UNIFORME.
 - v PREDECIBLE.
- v LA SOLUCION Y ESTUDIO DE UN SISTEMA DE COLAS SE SIMPLIFICA MUCHO SI SE SABE QUE SE ENCUENTRA EN ESTADO ESTABLE:
 - v CIERTOS PARAMETROS IMPORTANTES PERMANECEN FIJOS.
 - v RESULTA RELATIVAMENTE FACIL CATEGORIZAR LA OPERACION DEL SISTEMA.
- v LAS *SOLUCIONES TRANSITORIAS O DEPENDIENTES DEL TIEMPO*:
 - v SON MUCHO MAS COMPLEJAS.
 - v ESTAN FUERA DEL ALCANCE DE ESTE CURSO.
- v **RESULTADO DE LITTLE.**
- v ES UNA DE LAS MEDICIONES MAS SENCILLAS Y UTILES DEL RENDIMIENTO DE UN SISTEMA DE COLAS.
- v RELACIONA LAS SIGUIENTES CANTIDADES:
 - v W_q : TIEMPO MEDIO QUE EMPLEA UN CLIENTE EN UNA COLA.
 - v λ : TASA DE LLEGADAS.
 - v L_q : N° DE CLIENTES EN LA COLA.
 - v W : TIEMPO MEDIO QUE EMPLEA UN CLIENTE EN EL SISTEMA.
 - v L : N° DE CLIENTES EN EL SISTEMA.
- v EL RESULTADO DE LITTLE SE EXPRESA COMO:
 - v $L_q = \lambda W_q$
 - v $L = \lambda W$

RESUMEN DEL PROCESO DE POISSON

- v SE DEFINE “P(k,t)” COMO LA PROBABILIDAD DE EXACTAMENTE “k” LLEGADAS EN UN INTERVALO DE TIEMPO DE LONGITUD “t”.
- v UN PROCESO ES DE POISSON SI Y SOLO SI:
 - u PARA INTERVALOS APROPIADAMENTE PEQUEÑOS Δt :
 - Φ $P(k,t) = :$
 - $\lambda \Delta t$ PARA $k = 1$ (λ ES LA TASA PROMEDIO DE LLEGADAS.
 - $1 - \lambda \Delta t$ PARA $k = 0$.
 - 0 PARA $k > 1$.
 - u CUALESQUIERA EVENTOS DEFINIDOS PARA TENER LUGAR EN INTERVALOS DE TIEMPO NO SUPERPUESTOS SON MUTUAMENTE INDEPENDIENTES.
- v UN PROCESO TAMBIEN ES DE POISSON SI LOS TIEMPOS ENTRE LLEGADAS SUCESIVAS (*TIEMPOS ENTRE LLEGADAS DE PRIMER ORDEN*):
 - u SON VARIABLES ALEATORIAS EXPONENCIALES.
 - u IDENTICAMENTE DISTRIBUIDAS.
- v SI LA VARIABLE ALEATORIA “k” INDICA EL N° DE LLEGADAS:
 - u LA *PROBABILIDAD* DE, EXACTAMENTE, “k” LLEGADAS EN UN INTERVALO DE LONGITUD “t” ES:
 - Φ $P(k,t) = [(\lambda t)^k e^{-\lambda t}] / k!$ $t \geq 0; \quad k = 0, 1, 2, \dots$
 - u EL *VALOR ESPERADO* O *VALOR MEDIO* DE k ES:
 - Φ $E(k) = \lambda t$.
 - u LA *VARIANZA* DE k ES:
 - Φ $(\sigma_k)^2 = \lambda t$.
- v LA SUMA DE DOS VARIABLES DE POISSON ALEATORIAS INDEPENDIENTES “x” E “y” TAMBIEN DESCRIBEN UN PROCESO DE POISSON:
 - u LOS VALORES ESPERADOS SON:
 - Φ $E(y) = \mu_2 = \lambda_2 t$.
 - Φ $E(x) = \mu_1 = \lambda_1 t$.
 - u LA PROBABILIDAD DE “k” LLEGADAS EN EL TIEMPO “t” ES:
 - Φ $P(k,t) = [(\lambda_1 t + \lambda_2 t)^k e^{-(\lambda_1 t + \lambda_2 t)}] / k!$ $t \geq 0; \quad k = 0, 1, 2, \dots$
 - Φ $P(k,t) = [(\mu_1 + \mu_2)^k e^{-(\mu_1 + \mu_2)}] / k!$
 - Φ $P(k,t) = [\mu_s^k e^{-\mu_s}] / k!$ $\mu_s = \mu_1 + \mu_2$
 - Φ $P(k,t) = [(\lambda_s t)^k e^{-\lambda_s t}] / k!$ $\lambda_s = \lambda_1 + \lambda_2$

RESUMEN DEL PROCESO DE POISSON

- v LA SUMA DE “n” PROCESOS DE POISSON INDEPENDIENTES RESULTA EN UN PROCESO DE POISSON CON UNA TASA DE LLEGADA:

$$v \quad \lambda = \sum_{i=1}^n \lambda_i$$

- v PARA UN PROCESO DE POISSON CON UNA TASA DE LLEGADA “ λ ” SE PUEDE FORMAR UN NUEVO PROCESO DE POISSON UTILIZANDO *BORRADAS ALEATORIAS INDEPENDIENTES*:

- v CADA LLEGADA AL PROCESO ORIGINAL:

- Φ SE ACEPTA AL NUEVO PROCESO CON PROBABILIDAD “P”.
- Φ SE RECHAZA CON PROBABILIDAD “1 - P”.

- v LA TASA DE LLEGADA DEL NUEVO PROCESO DERIVADO ES “ λP ”.

- v LA GENERALIZACION PARA LA DESCOMPOSICION DE UN PROCESO DE POISSON EN “n” PROCESOS DERIVADOS INDEPENDIENTES, CADA UNO CON UNA PROBABILIDAD ASOCIADA “ p_i ” RESULTA:

$$v \quad \lambda_n = p_n \lambda.$$

$$v \quad \sum_{i=1}^n p_i = 1.$$

$$v \quad \sum_{i=1}^n \lambda_i = \sum_{i=1}^n p_i \lambda = \lambda \sum_{i=1}^n p_i = \lambda.$$

- v EN UN PROCESO DE POISSON:

- v LA PROBABILIDAD DE QUE NO HAYA LLEGADAS EN UN INTERVALO DE LONGITUD “t” ES:

$$v \quad P(0,t) = [(\lambda t)^0 e^{-\lambda t}] / 0! = e^{-\lambda t}.$$

- v LA PROBABILIDAD DE UNA O MAS LLEGADAS EN UN INTERVALO DE LONGITUD “t” ES:

$$v \quad 1 - P(0,t) = 1 - e^{-\lambda t}.$$

RESUMEN DEL PROCESO DE POISSON

- v LA FUNCION DE DENSIDAD DE PROBABILIDAD PARA EL *TIEMPO ENTRE LLEGADAS DE PRIMER ORDEN (TIEMPO HASTA LA PRIMER LLEGADA)* ES:
 - v $f_t(t) = \lambda e^{-\lambda t} \quad (t \geq 0).$
- v EL VALOR ESPERADO “t” ES:
 - v $E(t) = 1 / \lambda.$
- v LA VARIANZA ES:
 - v $(\sigma_t)^2 = 1 / \lambda^2.$
- v LA FUNCION DE DENSIDAD DE PROBABILIDAD PARA EL *TIEMPO ENTRE LLEGADAS DE ORDEN R-ESIMO (TIEMPO HASTA LA R-ESIMA LLEGADA)* ES:
 - v $f_t(t) = (\lambda^r t^{r-1} e^{-\lambda t}) / (r - 1)! \quad (t \geq 0, r = 1, 2, \dots).$
- v EL VALOR ESPERADO “t” ES:
 - v $E(t) = r / \lambda.$
- v LA DESVIACION ESTANDAR ES:
 - v $(\sigma_t)^2 = r / \lambda^2.$
- v LAS INSTALACIONES DE SERVICIO PUEDEN PROPORCIONAR TIEMPOS DE SERVICIO EXPONENCIALES:
 - v LA PROBABILIDAD DE QUE EL TIEMPO DE SERVICIO SEA MENOR O IGUAL A “t” ES:
 - Φ $P(S \leq t) = 1 - e^{-\mu t} \quad (t \geq 0).$
 - v LA TASA PROMEDIO DE SERVICIO ES “μ”.
 - v EL TIEMPO PROMEDIO DE SERVICIO ES “1 / μ”.
 - v LA FUNCION DE DENSIDAD DE PROBABILIDAD PARA EL TIEMPO DE SERVICIO “t” ES:
 - Φ $f_t(t) = \mu e^{-\mu t} \quad (t \geq 0).$
 - v LA MEDIA DEL TIEMPO DE SERVICIO ES:
 - Φ $E(s) = 1 / \mu.$
 - v LA VARIANZA ES “1 / μ²”.
- v UN SERVIDOR QUE OPERA DE ESTA MANERA SE DENOMINA *SERVIDOR EXPONENCIAL*.

ANALISIS DE UN SISTEMA DE COLAS

M/M/1

v FORMULAS DE ESTADO PARA EL SISTEMA DE COLAS M/M/C:

v INTENSIDAD DE TRAFICO:

$$\Phi \quad u = \lambda / \mu = \lambda E(s).$$

v UTILIZACION DEL SERVIDOR:

$$\Phi \quad \rho = u / c.$$

v PROBABILIDAD DE QUE TODOS LOS SERVIDORES ESTEN EN USO, POR LO QUE UN CLIENTE QUE LLEGA DEBE ESPERAR:

$$\Phi \quad C(c,u) = [(u^c) / c!] / \{ [(u^c) / c!] + (1-\rho) [\sum_{n=0}^{c-1} (u^n) / n!] \}.$$

v TIEMPO PROMEDIO EN LA COLA:

$$\Phi \quad W_q = [C(c,u) E(s)] / [c (1 - \rho)].$$

v TIEMPO PROMEDIO EN EL SISTEMA:

$$\Phi \quad W = W_q + E(s).$$

v PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA:

$$\Phi \quad \pi_q(90) = \{ [E(s)] / [c (1 - \rho)] \} \{ \ln[10 C(c,u)] \}.$$

v FORMULAS DE ESTADO PARA EL SISTEMA DE COLAS M/M/1:

v SE DEDUCEN DE LAS ANTERIORES:

$$\Phi \quad C(c,u) = \rho = \lambda E(s).$$

$$\Phi \quad W_q = [\rho E(s)] / (1 - \rho).$$

$$\Phi \quad W = E(s) / (1 - \rho).$$

$$\Phi \quad \pi_q(90) = W [\ln(10 \rho)].$$

v EJEMPLO PARA EL ANALISIS:

v LOS OPERADORES DE UNA EMPRESA PRECISAN USAR UN EQUIPO ESPECIAL.

v LA EMPRESA OPERA LAS 24 HS. DEL DIA.

v LOS 48 OPERADORES (COMO PROMEDIO) NECESITAN USAR EL EQUIPO UNA VEZ AL DIA.

v LOS OPERADORES LLEGAN AL EQUIPO EN FORMA ALEATORIA (LLEGADAS DE POISSON).

v EL TIEMPO QUE CADA OPERADOR UTILIZA EL EQUIPO ES EXPONENCIAL Y COMO PROMEDIO ES DE 20 MINUTOS.

ANALISIS DE UN SISTEMA DE COLAS

M/M/1

- v UTILIZANDO UN SISTEMA DE COLAS M/M/1 PARA MODELAR EL USO DEL EQUIPO ESPECIAL DEL EJEMPLO SE OBTIENE:
 - u UTILIZACION DEL EQUIPO:
 - Φ $u = \lambda E(s) = (48 / 24) \cdot (1 / 3) = 2 / 3$; $\rho = 2 / 3$; $E(s) = 20$ MINUTOS.
 - u TIEMPO PROMEDIO DE ESPERA DE UN OPERADOR ANTES DE USAR EL EQUIPO:
 - Φ $W_q = [\rho E(s)] / (1 - \rho) = [(2 / 3) 20] / (1 / 3) = 40$ MINUTOS.
 - u TIEMPO TOTAL QUE UN OPERADOR UTILIZA EL EQUIPO:
 - Φ $W = W_q + E(s) = 40 \text{ MIN.} + 20 \text{ MIN.} = 60$ MINUTOS.
 - u PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA:
 - Φ $\pi_q(90) = W [\ln(10 \rho)] = 60 \ln(6,667) = 113,826$ MINUTOS:
 - UN 10% DE LOS OPERADORES (UNOS 5 POR DIA) SUFRE PROLONGADAS ESPERAS DE CASI 2 HORAS.
- v SEGUN EL RESULTADO DE LITTLE:
 - u TASA DE LLEGADA DE OPERADORES AL EQUIPO:
 - Φ $\lambda = 48 / 24 (60) = 1 / 30$ OPERADORES POR MINUTO.
 - u OPERADORES EN ESPERA:
 - Φ $L_q = (1 / 30) \cdot 40 = 1,33$ OPERADORES EN ESPERA.
 - u OPERADORES EN EL SISTEMA:
 - Φ $L = (1 / 30) \cdot 60 = 2$ OPERADORES EN EL CUARTO DEL EQUIPO.
- v CONCLUSION:
 - u *UN SOLO EQUIPO NO ES SUFICIENTE PARA HACER FRENTE A LAS NECESIDADES DE LOS OPERADORES SIN PROVOCAR ESPERAS EXCESIVAS.*

ANALISIS DE UN SISTEMA DE COLAS

M/M/C

v EJEMPLO PARA EL ANALISIS:

- u EL MISMO DEL TEMA ANTERIOR.
- u SI LA DECISION ES ADQUIRIR MAS EQUIPOS, LOS INTERROGANTES SON LOS SIGUIENTES:
 - ⊕ ¿CUANTOS EQUIPOS ADICIONALES DEBERAN ADQUIRIRSE PARA MANTENER EL PERCENTIL 90 DE TIEMPO EN ESPERA POR DEBAJO DE 10 MINUTOS?.
 - ⊕ ¿DEBERAN MANTENERSE TODOS LOS EQUIPOS EN UN LUGAR CENTRAL, O DEBERAN DISTRIBUIRSE POR TODO EL EDIFICIO?.
 - (NOTA: SE DEBE IGNORAR EL TIEMPO QUE LES LLEVA A LOS OPERADORES LLEGAR HASTA LOS EQUIPOS).

v COLOCANDO LOS EQUIPOS EN DIFERENTES LUGARES DE LA EMPRESA:

- u C / U DE LOS EQUIPOS SE DEBE ANALIZAR COMO UN SISTEMA DE COLAS M/M/1.
- u LA CARGA DE TRABAJO DEBE DIVIDIRSE EN PARTES IGUALES ENTRE LOS EQUIPOS.
- u COLOCANDO 1, 2, 3, 4 O 5 EQUIPOS EN LOCALIDADES SEPARADAS OBTENEMOS LOS SIGUIENTES VALORES:
 - ⊕ UTILIZACION DEL SERVIDOR: ρ : 2/3; 1/3; 2/9; 1/6 Y 2/15.
 - ⊕ TIEMPO DE ESPERA DE SERVICIO: $E(s)$: 20 MINUTOS EN TODOS LOS CASOS.
 - ⊕ TIEMPO DE ESPERA EN LA COLA: W_q : 40; 10; 5,7; 4 Y 3,1 MINUTOS.
 - ⊕ TIEMPO DE ESPERA EN EL SISTEMA: W : 60; 30; 25,7; 24 Y 23 MINUTOS.
 - ⊕ PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA: $\pi_q(90)$: 113,8; 36,1; 20,5; 12,3 Y 6,6 MINUTOS.
- u CONCLUSIONES:
 - ⊕ *LOS TIEMPOS DE ESPERA EN LA COLA BAJAN MUY RAPIDO TAN PRONTO COMO SE AÑADE EL SEGUNDO EQUIPO M/M/1.*
 - ⊕ *EL PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA ES SUPERIOR A 10 MIN. HASTA QUE SE AÑADE EL QUINTO EQUIPO.*

ANALISIS DE UN SISTEMA DE COLAS

M/M/C

- v COLOCANDO TODOS LOS EQUIPOS EN UN LUGAR CENTRAL:
 - u SE CONSIDERA UN SISTEMA DE COLAS M/M/2 SENCILLO.
 - u UTILIZANDO LAS FORMULAR DE LOS SISTEMAS M/M/C SE OBTIENEN LOS SIGUIENTES VALORES:
 - Φ INTENSIDAD DE TRAFICO: u : 2/3.
 - Φ UTILIZACION DEL SERVIDOR: ρ : 1/3.
 - Φ PROBABILIDAD DE QUE TODOS LOS SERVIDORES SE ENCUENTREN EN ESTE MOMENTO EN USO, POR LO QUE UN CLIENTE QUE LLEGA DEBE ESPERAR: $C(c,u)$: 1/6.
 - Φ TIEMPO PROMEDIO EN LA COLA: W_q : 2,5 MINUTOS.
 - Φ TIEMPO PROMEDIO EN EL SISTEMA: W : 22,5 MINUTOS.
 - Φ PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA: $\pi_q(90)$: 7,66 MINUTOS.
 - u CONCLUSIONES:
 - Φ EL PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA DEL SISTEMA M/M/2 ES INFERIOR AL CRITERIO DE 10 MINUTOS.
 - Φ CON SOLO 2 EQUIPOS CENTRALIZADOS EN UNA POSICION SE PUEDEN ELIMINAR LOS PROBLEMAS DE ESPERA DEL SISTEMA DE UN SOLO EQUIPO.
 - Φ PARA ASEGURAR UN PERCENTIL 90 DE TIEMPO DE ESPERA EN LA COLA INFERIOR A 10 MINUTOS SERAN NECESARIOS:
 - 5 EQUIPOS M/M/1 DISTRIBUIDOS, O.
 - 2 EQUIPOS EN UNA CONFIGURACION M/M/2 CENTRAL.

PROCESOS DE MARKOV

- v UN *PROCESO DE MARKOV* ES UN MODELO ADECUADO PARA DESCRIBIR EL COMPORTAMIENTO DE SISTEMAS DONDE:
 - υ EL SISTEMA ESTA SITUADO EN UNO DE UN CONJUNTO DE *ESTADOS DISCRETOS MUTUAMENTE EXCLUYENTES Y COLECTIVAMENTE EXHAUSTIVOS* $S_0, S_1, S_2, \dots, S_n$.
- v EL *ESTADO PRESENTE* DEL SISTEMA Y LAS *PROBABILIDADES DE TRANSICION* ENTRE VARIOS ESTADOS DEL SISTEMA:
 - υ CARACTERIZAN EL COMPORTAMIENTO FUTURO DEL SISTEMA.
- v DADO QUE UN *PROCESO DE MARKOV* SE ENCUENTRA EN UN ESTADO DETERMINADO, SU COMPORTAMIENTO FUTURO NO DEPENDE DE SU HISTORIA ANTERIOR A SU ENTRADA A ESE ESTADO.
- v MUCHOS PROCESOS DE MARKOV EXHIBEN UN COMPORTAMIENTO DE ESTADO ESTABLE:
 - υ LAS PROBABILIDADES DE QUE EL PROCESO SE ENCUENTRE EN UN ESTADO DETERMINADO SON CONSTANTES EN EL TIEMPO.
- v SE DICE QUE UN ESTADO " S_j " ES *TRANSITORIO* SI DESDE UN ESTADO " S_k " QUE PUEDE SER ALCANZADO DESDE " S_j ", EL SISTEMA NO PUEDE REGRESAR A " S_j ".
- v SE DICE QUE UN ESTADO " S_j " ES *RECURRENTE* SI DESDE CADA ESTADO " S_k " ALCANZABLE DESDE " S_j ", EL SISTEMA PUEDE REGRESAR A " S_k ".
- v UNA *CADENA SENCILLA* ES UNA SERIE DE ESTADOS RECURRENTES TAL QUE EL SISTEMA PUEDE LLEGAR A CUALQUIER ESTADO DE LA CADENA DESDE CUALQUIER OTRO ESTADO DE ESTA.
- v UN CAMBIO DE ESTADO EN UN *PROCESO DE MARKOV DE TRANSICION CONTINUA* PUEDE PRODUCIR CAMBIOS DE ESTADO EN CUALQUIER INSTANTE DE UNA ESCALA DE TIEMPO CONTINUA.

PROCESOS DE NACIMIENTO Y MUERTE

- v SON UN CASO IMPORTANTE DE LOS PROCESOS DE MARKOV.
- v SON PARTICULARMENTE APLICABLES AL MODELADO DE SISTEMAS DE COMPUTACION.
- v UN *PROCESO DE MARKOV DE NACIMIENTO Y MUERTE CONTINUO* TIENE LA PROPIEDAD DE QUE:
 - v $\lambda_{ij} = 0$ SI $j \neq i + 1$ Y $j \neq i - 1$.
 - v λ_{ij} ES LA TASA A LA CUAL OCURREN LAS TRANSICIONES DEL ESTADO “ S_i ” AL ESTADO “ S_j ”.
 - v $\lambda_{i(i+1)} = b_i$ ES LA *TASA PROMEDIO DE NACIMIENTO* DESDE EL ESTADO “ S_i ”.
 - v $\lambda_{i(i-1)} = d_i$ ES LA *TASA PROMEDIO DE MUERTE* DESDE EL ESTADO “ S_i ”.
 - v “ P_i ” ES LA *PROBABILIDAD DE ESTADO ESTABLE* DE QUE EL PROCESO SE ENCUENTRE EN EL ESTADO “ S_i ”.
- v EN ESTADO ESTABLE, EN CUALQUIER INTERVALO DE TIEMPO ALEATORIO “ Δt ”, EL PROCESO PUEDE REALIZAR LAS SIGUIENTES TRANSICIONES CON LA MISMA PROBABILIDAD:
 - v $S_i \rightarrow S_{i+1}$ CON UNA PROBABILIDAD $P_i b_i$.
 - v $S_{i+1} \rightarrow S_i$ CON UNA PROBABILIDAD $P_{i+1} d_{i+1}$.
 - v $P_i b_i = P_{i+1} d_{i+1}$.
- v LA RESOLUCION DE UN PROCESO DE NACIMIENTO Y MUERTE CONTINUO SIGNIFICA DETERMINAR LOS DIFERENTES “ P_i ” USANDO LAS RELACIONES:
 - v $P_{i+1} = (b_i / d_{i+1}) P_i$.
 - v $\sum_i P_i = 1$.

ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO

- v SE SUPONE LA SIGUIENTE SITUACION:
 - u LAS PETICIONES DE ACCESO A DISCO LLEGAN COMO UN PROCESO DE POISSON CON UNA TASA PROMEDIO DE λ PETICIONES POR MINUTO.
 - u SI EL DISCO ESTA EN USO, LA PETICION SE COLOCA EN UNA COLA PRIMERO EN LLEGAR, PRIMERO EN SER SERVIDO.
 - u CUANDO EL DISCO QUEDA DISPONIBLE SE SIRVE LA PRIMERA PETICION DE LA COLA.
 - u EL TIEMPO DE SERVICIO ES UNA VARIABLE ALEATORIA EXPONENCIALMENTE DISTRIBUIDA CON UN VALOR ESPERADO DE $1 / \mu$ MINUTOS.
 - u LA TASA PROMEDIO DE SERVICIO ES DE μ PETICIONES POR MINUTO.
- v SE DEBE DETERMINAR, PARA C / U DE LOS CASOS:
 - u EL VALOR ESPERADO PARA EL N° TOTAL DE PETICIONES AL DISCO PENDIENTES (EN LA COLA O EN SERVICIO).
 - u LAS PROBABILIDADES DEL ESTADO LIMITE.
- v CASO I:
 - u EL DISPOSITIVO DE DISCO CONTIENE UN SOLO BRAZO.
 - u SOLO PUEDE DAR SERVICIO A UNA PETICION A LA VEZ.
 - u LA TASA DE SERVICIO ES μ .
- v CASO II:
 - u EL DISPOSITIVO DE DISCO CONTIENE GRAN N° DE BRAZOS MOVILES.
 - u CADA BRAZO PUEDE DAR SERVICIO A UNA PETICION DE DISCO A LA MISMA TASA μ .
 - u SE SUPONE QUE UN N° INFINITO DE PETICIONES PUEDEN RECIBIR SERVICIO EN PARALELO.

ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO

- v *SOLUCION AL CASO I.*
- v S_i ES EL ESTADO DEL SISTEMA CUANDO HAY i PETICIONES DE DISCO AL DISPOSITIVO DE SERVICIO DE DISCO.
- v LA TASA DE LLEGADAS DE PETICIONES ES INDEPENDIENTE DEL ESTADO DEL SISTEMA:
 - v LA PROBABILIDAD DE LA TRANSICION $S_i \rightarrow S_{i+1}$ EN EL SIGUIENTE INTERVALO DE TIEMPO Δt ES $\lambda \Delta t$.
- v SE CONSIDERA AL SISTEMA COMO UN PROCESO DE NACIMIENTO Y MUERTE CONTINUO DE CADENA SENCILLA Y ESTADOS INFINITOS CON:
 - v $d_i = 0$ $i = 0.$
 - v $d_i = \mu$ $i = 1, 2, 3, \dots$
 - v $b_i = \lambda$ $i = 0, 1, 2, \dots$
 - v SOLO UNA PETICION PUEDE SER SERVIDA EN UN MOMENTO DADO Y SE SIRVE A UNA TASA μ .
 - v $\mu > \lambda$:
 - Φ ASEGURA QUE LA LONGITUD DE LA COLA DE PETICIONES EN ESPERA NO CREZCA INDEFINIDAMENTE.
- v SE UTILIZAN LAS RELACIONES:
 - v $P_{i+1} = (b_i / d_{i+1}) P_i$. $i = 0, 1, 2, \dots$
 - v $\sum_i P_i = 1$.
 - v $P_1 = (\lambda / \mu) P_0$.
 - v $P_2 = (\lambda / \mu) P_1 = (\lambda / \mu)^2 P_0$.
 - v $P_i = (\lambda / \mu)^i P_0$.
 - v $\sum_i P_i = 1 = \sum_i (\lambda / \mu)^i P_0 = 1 / [1 - (\lambda / \mu)] P_0$.
 - v $P_0 = 1 - (\lambda / \mu)$: *PROBABILIDAD DE QUE EL SISTEMA SE ENCUENTRE OCIOSO.*
 - v $P_i = (\lambda / \mu)^i P_0 = [1 - (\lambda / \mu)] (\lambda / \mu)^i$. $i = 0, 1, 2, \dots$
 - v $P_i = [1 - (\lambda / \mu)] (\lambda / \mu)^i$: *PROBABILIDAD QUE HAYAN i PETICIONES PENDIENTES.*

ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO

v *EL N° PROMEDIO DE PETICIONES PENDIENTES ES:*

$$v \quad E(i) = \sum_i i P_i = \sum_i i [1 - (\lambda / \mu)] (\lambda / \mu)^i = [1 - (\lambda / \mu)] \sum_i i (\lambda / \mu)^i =$$

$$v \quad E(i) = [1 - (\lambda / \mu)] (\lambda / \mu) \sum_i i (\lambda / \mu)^{i-1} =$$

$$v \quad E(i) = [1 - (\lambda / \mu)] (\lambda / \mu) \{1 / [1 - (\lambda / \mu)^2]\} =$$

$$v \quad E(i) = (\lambda / \mu) [1 - (\lambda / \mu)^{-1}].$$

v *SOLUCION AL CASO II:*

v CON i PETICIONES SIENDO SERVIDAS:

v LA PROBABILIDAD DE QUE UNA PETICION EN PARTICULAR ACABE SIENDO SERVIDA DENTRO DEL SIGUIENTE Δt ES $\mu \Delta t$.

v LA PROBABILIDAD DE QUE EXACTAMENTE UNA PETICION CUALQUIERA ACABE ES $i \mu \Delta t$ (BUENA APROXIMACION DE PRIMER ORDEN).

v CUALQUIERA DE LAS i PETICIONES PUEDE TERMINAR Y PROVOCAR UN CAMBIO DE ESTADO.

v EL SISTEMA SE VE COMO UN PROCESO DE NACIMIENTO Y MUERTE CONTINUO DE CADENA SENCILLA Y DE ESTADOS INFINITOS CON:

$$v \quad b_i = \lambda \quad i = 0, 1, 2, \dots$$

$$v \quad d_i = 0 \quad i = 0.$$

$$v \quad d_i = i \mu \quad i = 1, 2, 3, \dots$$

v NINGUN CLIENTE TIENE QUE ESPERAR YA QUE SE SUPONEN INFINITOS SERVIDORES EN PARALELO.

v SE UTILIZAN LAS RELACIONES:

$$v \quad P_{i+1} = (b_i / d_{i+1}) P_i. \quad i = 0, 1, 2, \dots$$

$$v \quad \sum_i P_i = 1.$$

$$v \quad P_1 = (\lambda / \mu) P_0.$$

$$v \quad P_2 = (\lambda / 2\mu) P_1 = (1 / 2) (\lambda / \mu)^2 P_0.$$

$$v \quad P_3 = (\lambda / 3\mu) P_2 = (1 / (3 \cdot 2)) (\lambda / \mu)^3 P_0.$$

$$v \quad P_i = (1 / i!) (\lambda / \mu)^i P_0.$$

$$v \quad \sum_i P_i = 1 = \sum_i (1 / i!) (\lambda / \mu)^i P_0.$$

ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO

- $$\sum_n (x^n / n!) = e^x.$$
- $$\sum_i P_i = 1 = \sum_i (1 / i!) (\lambda / \mu)^i P_0 = e^{\lambda / \mu} P_0.$$
- $$P_0 = e^{-\lambda / \mu}.$$
- $$P_i = (\lambda / \mu)^i [(e^{-\lambda / \mu}) / i!].$$
- $$E(i) = \sum_i i P_i = \sum_i i (\lambda / \mu)^i [(e^{-\lambda / \mu}) / i!] = (e^{-\lambda / \mu}) \sum_i i (\lambda / \mu)^i (1 / i!) =$$
- $$E(i) = (e^{-\lambda / \mu}) \sum_i (\lambda / \mu) (\lambda / \mu)^{i-1} [1 / (i-1)!] =$$
- $$E(i) = (e^{-\lambda / \mu}) (\lambda / \mu) \sum_i [1 / (i-1)!] (\lambda / \mu)^{i-1} =$$
- $$E(i) = (e^{-\lambda / \mu}) (\lambda / \mu) (e^{\lambda / \mu}) =$$
- $$E(i) = (\lambda / \mu).$$

CONCLUSIONES:

- EN EL SISTEMA DE UN SOLO SERVIDOR, SI UNA PETICION QUE LLEGA ENCUENTRA OCUPADO EL DISPOSITIVO DE DISCO:
 - Φ DEBE ESPERAR.
- EN EL SISTEMA DE SERVIDORES INFINITOS, LAS PETICIONES QUE LLEGAN:
 - Φ SIEMPRE ENTRAN AL SERVICIO DE INMEDIATO.
- EN EL SISTEMA DE UN SOLO SERVIDOR:
 - Φ A MEDIDA QUE λ TIENDE A μ LA PROBABILIDAD DE QUE EL SISTEMA SE ENCUENTRE OCIOSO DECRECE RAPIDAMENTE:
 - \bullet LAS PETICIONES QUE LLEGAN ESPERAN.
 - Φ EL N° PROMEDIO DE PETICIONES PENDIENTES CRECE CON RAPIDEZ.
- EN EL SISTEMA DE SERVIDORES INFINITOS:
 - Φ EL N° PROMEDIO DE PETICIONES PENDIENTES TIENDE A 1.

SEGURIDAD DE LOS SISTEMAS OPERATIVOS

- v INTRODUCCION A LA SEGURIDAD DE LOS SISTEMAS OPERATIVOS
- v REQUISITOS DE SEGURIDAD
- v UN TRATAMIENTO TOTAL DE LA SEGURIDAD
- v SEGURIDAD EXTERNA Y SEGURIDAD OPERACIONAL
- v VIGILANCIA, VERIFICACION DE AMENAZAS Y AMPLIFICACION
- v PROTECCION POR CONTRASEÑA
- v AUDITORIA Y CONTROLES DE ACCESO
- v NUCLEOS DE SEGURIDAD Y SEGURIDAD POR HARDWARE
- v SISTEMAS SUPERVIVIENTES
- v CAPACIDADES Y SISTEMAS ORIENTADOS HACIA EL OBJETO
- v CRIPTOGRAFIA
- v PENETRACION AL SISTEMA OPERATIVO

INTRODUCCION A LA SEGURIDAD DE LOS SISTEMAS OPERATIVOS

- v LA EVOLUCION DE LA COMPUTACION Y DE LAS COMUNICACIONES EN LAS ULTIMAS DECADAS:
 - u HA HECHO MAS ACCESIBLES A LOS SISTEMAS INFORMATICOS.
 - u HA INCREMENTADO LOS RIESGOS VINCULADOS A LA SEGURIDAD.
- v LA VULNERABILIDAD DE LAS COMUNICACIONES DE DATOS ES UN ASPECTO CLAVE DE LA SEGURIDAD DE LOS SISTEMAS INFORMATICOS:
 - u LA IMPORTANCIA DE ESTE ASPECTO ES CADA VEZ MAYOR EN FUNCION DE LA PROLIFERACION DE LAS REDES DE COMPUTADORAS.
- v EL NIVEL DE CRITICIDAD Y DE CONFIDENCIALIDAD DE LOS DATOS ADMINISTRADOS POR LOS SISTEMAS INFORMATICOS ES CADA VEZ MAYOR:
 - u EJ.: CORREO PERSONAL, TRANSFERENCIA DE FONDOS, CONTROL DE MANUFACTURA, CONTROL DE SISTEMAS DE ARMAS, CONTROL DE TRAFICO AEREO, CONTROL DE IMPLANTES MEDICOS (MARCAPASOS, ETC.).
 - u LOS SISTEMAS DEBEN FUNCIONAR ININTERRUMPIDAMENTE Y SIN PROBLEMAS.
- v EL SISTEMA OPERATIVO, COMO ADMINISTRADOR DE LOS RECURSOS DEL SISTEMA:
 - u CUMPLE UNA FUNCION MUY IMPORTANTE EN LA INSTRUMENTACION DE LA SEGURIDAD.
 - u NO ENGLOBA A TODOS LOS ASPECTOS DE LA SEGURIDAD.
 - u DEBE SER COMPLEMENTADO CON MEDIDAS EXTERNAS AL S. O.
- v LA SIMPLE SEGURIDAD FISICA RESULTA INSUFICIENTE ANTE LA POSIBILIDAD DE ACCESO MEDIANTE EQUIPOS REMOTOS CONECTADOS.
- v LA TENDENCIA ES QUE LOS SISTEMAS SEAN MAS ASEQUIBLES Y FACILES DE USAR:
 - u LA FAVORABILIDAD HACIA EL USUARIO PUEDE IMPLICAR UN AUMENTO DE LA VULNERABILIDAD.
- v SE DEBEN IDENTIFICAR LAS AMENAZAS POTENCIALES:
 - u PUEDEN PROCEDER DE FUENTES MALICIOSAS O NO.
- v EL NIVEL DE SEGURIDAD A PROPORCIONAR DEPENDE DEL VALOR DE LOS RECURSOS QUE HAY QUE ASEGURAR.

REQUISITOS DE SEGURIDAD

UN TRATAMIENTO TOTAL DE LA SEGURIDAD

- v **REQUISITOS DE SEGURIDAD.**
- v LOS REQUISITOS DE SEGURIDAD DE UN SISTEMA DADO DEFINEN LO QUE SIGNIFICA LA SEGURIDAD, PARA ESE SISTEMA.
- v LOS REQUISITOS SIRVEN DE BASE PARA DETERMINAR SI EL SISTEMA IMPLEMENTADO ES SEGURO:
 - υ SIN UNA SERIE DE REQUISITOS PRECISOS TIENE POCO SENTIDO CUESTIONAR LA SEGURIDAD DE UN SISTEMA.
 - υ SI LOS REQUISITOS ESTAN DEBILMENTE ESTABLECIDOS NO DICEN MUCHO SOBRE LA VERDADERA SEGURIDAD DEL SISTEMA.
- v ALGUNOS EJEMPLOS DE FORMULACION DE LOS REQUISITOS DE SEGURIDAD SON LOS SIGUIENTES:
 - υ *DIRECTIVA DOD 5200.28 (EE.UU.):*
 - ⊕ ESPECIFICA COMO DEBE MANIPULARSE LA INFORMACION CLASIFICADA EN SISTEMAS DE PROCESAMIENTO DE DATOS.
 - υ *MANUAL DE REFERENCIA DE TECNOLOGIA DE SEGURIDAD DE LA COMPUTADORA (EE. UU.):*
 - ⊕ ESPECIFICA COMO EVALUAR LA SEGURIDAD DE LOS SISTEMAS DE COMPUTACION DE LA FUERZA AEREA.
 - υ *LEY DE INTIMIDAD DE 1974 (EE. UU.):*
 - ⊕ REQUIERE QUE LAS AGENCIAS FEDERALES ASEGUEN LA INTEGRIDAD Y SEGURIDAD DE LA INFORMACION ACERCA DE LOS INDIVIDUOS:
 - ESPECIALMENTE EN EL CONTEXTO DEL AMPLIO USO DE LAS COMPUTADORAS.
- v **UN TRATAMIENTO TOTAL DE LA SEGURIDAD.**
- v UN TRATAMIENTO TOTAL INCLUYE ASPECTOS DE LA SEGURIDAD DEL COMPUTADOR DISTINTOS A LOS DE LA SEGURIDAD DE LOS S. O.
- v LA *SEGURIDAD EXTERNA* DEBE ASEGURAR LA INSTALACION COMPUTACIONAL CONTRA INTRUSOS Y DESASTRES COMO INCENDIOS E INUNDACIONES:
 - υ CONCEDIDO EL ACCESO FISICO EL S. O. DEBE IDENTIFICAR AL USUARIO ANTES DE PERMITIRLE EL ACCESO A LOS RECURSOS:
 - ⊕ *SEGURIDAD DE LA INTERFAZ DEL USUARIO.*
- v LA *SEGURIDAD INTERNA* TRATA DE LOS CONTROLES INCORPORADOS AL HARDWARE Y AL S. O. PARA ASEGURAR LA CONFIABILIDAD, OPERABILIDAD Y LA INTEGRIDAD DE LOS PROGRAMAS Y DATOS.

SEGURIDAD EXTERNA Y SEGURIDAD OPERACIONAL

- v **SEGURIDAD EXTERNA.**
- v LA SEGURIDAD EXTERNA CONSISTE EN:
 - υ *SEGURIDAD FISICA.*
 - υ *SEGURIDAD OPERACIONAL.*
- v LA SEGURIDAD FISICA INCLUYE:
 - υ *PROTECCION CONTRA DESASTRES.*
 - υ *PROTECCION CONTRA INTRUSOS.*
- v EN LA SEGURIDAD FISICA SON IMPORTANTES LOS *MECANISMOS DE DETECCION*, ALGUNOS EJEMPLOS SON:
 - υ DETECTORES DE HUMO.
 - υ SENSORES DE CALOR.
 - υ DETECTORES DE MOVIMIENTO.
- v LA PROTECCION CONTRA DESASTRES PUEDE SER COSTOSA Y FRECUENTEMENTE NO SE ANALIZA EN DETALLE:
 - υ DEPENDE EN GRAN MEDIDA DE LAS CONSECUENCIAS DE LA PERDIDA.
- v LA SEGURIDAD FISICA TRATA ESPECIALMENTE DE IMPEDIR LA ENTRADA DE INTRUSOS:
 - υ SE UTILIZAN SISTEMAS DE IDENTIFICACION FISICA:
 - Φ TARJETAS DE IDENTIFICACION.
 - Φ SISTEMAS DE HUELLAS DIGITALES.
 - Φ IDENTIFICACION POR MEDIO DE LA VOZ.

SEGURIDAD EXTERNA Y SEGURIDAD OPERACIONAL

- v **SEGURIDAD OPERACIONAL.**
- v CONSISTE EN LAS DIFERENTES POLITICAS Y PROCEDIMIENTOS IMPLEMENTADOS POR LA ADMINISTRACION DE LA INSTALACION COMPUTACIONAL.
- v LA *AUTORIZACION* DETERMINA QUE ACCESO SE PERMITE Y A QUIEN.
- v LA *CLASIFICACION* DIVIDE EL PROBLEMA EN SUBPROBLEMAS:
 - u LOS DATOS DEL SISTEMA Y LOS USUARIOS SE DIVIDEN EN CLASES:
 - ⊕ A LAS CLASES SE CONCEDEN DIFERENTES DERECHOS DE ACCESO.
- v UN ASPECTO CRITICO ES LA SELECCION Y ASIGNACION DE PERSONAL:
 - u LA PREGUNTA ES SI SE PUEDE CONFIAR EN LA GENTE.
 - u EL TRATAMIENTO QUE GENERALMENTE SE DA AL PROBLEMA ES LA *DIVISION DE RESPONSABILIDADES*:
 - ⊕ SE OTORGAN DISTINTOS CONJUNTOS DE RESPONSABILIDADES.
 - ⊕ NO ES NECESARIO QUE SE CONOZCA LA TOTALIDAD DEL SISTEMA PARA CUMPLIR CON ESAS RESPONSABILIDADES.
 - ⊕ PARA PODER COMPROMETER AL SISTEMA PUEDE SER NECESARIA LA COOPERACION ENTRE MUCHAS PERSONAS:
 - SE REDUCE LA PROBABILIDAD DE VIOLAR LA SEGURIDAD.
 - ⊕ DEBE INSTRUMENTARSE UN GRAN NUMERO DE VERIFICACIONES Y BALANCES EN EL SISTEMA PARA AYUDAR A LA DETECCION DE BRECHAS EN LA SEGURIDAD.
 - ⊕ EL PERSONAL DEBE ESTAR AL TANTO DE QUE EL SISTEMA DISPONE DE CONTROLES, PERO:
 - DEBE DESCONOCER CUALES SON ESOS CONTROLES:
 - SE REDUCE LA PROBABILIDAD DE PODER EVITARLOS.
 - DEBE PRODUCIRSE UN EFECTO DISUASIVO RESPECTO DE POSIBLES INTENTOS DE VIOLAR LA SEGURIDAD.
- v PARA DISEÑAR MEDIDAS EFECTIVAS DE SEGURIDAD SE DEBE PRIMERO:
 - u ENUMERAR Y COMPRENDER LAS AMENAZAS POTENCIALES.
 - u DEFINIR QUE GRADO DE SEGURIDAD SE DESEA (Y CUANTO SE ESTA DISPUESTO A GASTAR EN SEGURIDAD).
 - u ANALIZAR LAS CONTRAMEDIDAS DISPONIBLES.

VIGILANCIA, VERIFICACION DE AMENAZAS Y AMPLIFICACION

v **VIGILANCIA.**

v LA VIGILANCIA TIENE QUE VER CON:

- u LA VERIFICACION Y LA AUDITORIA DEL SISTEMA.
- u LA AUTENTIFICACION DE LOS USUARIOS.

v LOS SISTEMAS SOFISTICADOS DE AUTENTIFICACION DE USUARIOS RESULTAN MUY DIFICILES DE EVITAR POR PARTE DE LOS INTRUSOS.

v UN PROBLEMA EXISTENTES ES LA POSIBILIDAD DE QUE EL SISTEMA RECHACE A USUARIOS LEGITIMOS:

- u UN SISTEMA DE RECONOCIMIENTO DE VOZ PODRIA RECHAZAR A UN USUARIO LEGITIMO RESFRIADO.
- u UN SISTEMA DE HUELLAS DIGITALES PODRIA RECHAZAR A UN USUARIO LEGITIMO QUE TENGA UNA CORTADURA O UNA QUEMADURA.

v **VERIFICACION DE AMENAZAS.**

v ES UNA TECNICA SEGUN LA CUAL LOS USUARIOS NO PUEDEN TENER ACCESO DIRECTO A UN RECURSO:

- u SOLO LO TIENEN LAS RUTINAS DEL S. O. LLAMADAS *PROGRAMAS DE VIGILANCIA*.
- u EL USUARIO SOLICITA EL ACCESO AL S. O.
- u EL S. O. NIEGA O PERMITE EL ACCESO.
- u EL ACCESO LO HACE UN PROGRAMA DE VIGILANCIA QUE LUEGO PASA LOS RESULTADOS AL PROGRAMA DEL USUARIO.
- u PERMITE:
 - ⊕ DETECTAR LOS INTENTOS DE PENETRACION EN EL MOMENTO EN QUE SE PRODUCEN.
 - ⊕ ADVERTIR EN CONSECUENCIA.

v **AMPLIFICACION.**

v LA AMPLIFICACION SE PRODUCE CUANDO:

- u UN PROGRAMA DE VIGILANCIA NECESITA PARA CUMPLIR SU COMETIDO MAYORES DERECHOS DE ACCESO DE LOS QUE DISPONEN LOS USUARIOS:
 - ⊕ EJ.: SE REQUIERE CALCULAR UN PROMEDIO PARA LO CUAL ES NECESARIO LEER UN CONJUNTO DE REGISTROS A LOS QUE EL USUARIO NO TIENE ACCESO INDIVIDUALMENTE.

PROTECCION POR CONTRASEÑA

- v LAS CLASES DE ELEMENTOS DE *AUTENTIFICACION* PARA ESTABLECER LA IDENTIDAD DE UNA PERSONA SON:
 - v *ALGO SOBRE LA PERSONA:*
 - ⊕ EJ.: HUELLAS DIGITALES, REGISTRO DE LA VOZ, FOTOGRAFIA, FIRMA, ETC.
 - v *ALGO POSEIDO POR LA PERSONA:*
 - ⊕ EJ.: INSIGNIAS ESPECIALES, TARJETAS DE IDENTIFICACION, LLAVES, ETC.
 - v *ALGO CONOCIDO POR LA PERSONA:*
 - ⊕ EJ.: CONTRASEÑAS, COMBINACIONES DE CERRADURAS, ETC.
- v EL ESQUEMA MAS COMUN DE AUTENTIFICACION ES LA *PROTECCION POR CONTRASEÑA*:
 - v EL USUARIO ELIGE UNA PALABRA CLAVE, LA MEMORIZA, LA TECLEA PARA SER ADMITIDO EN EL SISTEMA COMPUTARIZADO:
 - ⊕ LA CLAVE NO DEBE DESPLEGARSE EN PANTALLA NI APARECER IMPRESA.
- v LA PROTECCION POR CONTRASEÑAS TIENE CIERTAS DESVENTAJAS SI NO SE UTILIZAN CRITERIOS ADECUADOS PARA:
 - v ELEGIR LAS CONTRASEÑAS.
 - v COMUNICARLAS FEHACIENTEMENTE EN CASO DE QUE SEA NECESARIO.
 - v DESTRUIR LAS CONTRASEÑAS LUEGO DE QUE HAN SIDO COMUNICADAS.
 - v MODIFICARLAS LUEGO DE ALGUN TIEMPO.
- v LOS USUARIOS TIENDEN A ELEGIR CONTRASEÑAS FACILES DE RECORDAR:
 - v NOMBRE DE UN AMIGO, PARIENTE, PERRO, GATO, ETC.
 - v N° DE DOCUMENTO, DOMICILIO, PATENTE DEL AUTO, ETC.
- v ESTOS DATOS PODRIAN SER CONOCIDOS POR QUIEN INTENTE UNA VIOLACION A LA SEGURIDAD MEDIANTE *INTENTOS REPETIDOS*:
 - v DEBE LIMITARSE LA CANTIDAD DE INTENTOS FALLIDOS DE ACIERTO PARA EL INGRESO DE LA CONTRASEÑA.
- v LA CONTRASEÑA NO DEBE SER MUY CORTA PARA NO FACILITAR LA PROBABILIDAD DE ACIERTO.
- v TAMPOCO DEBE SER MUY LARGA PARA QUE NO SE DIFICULTE SU MEMORIZACION:
 - v LOS USUARIOS LA ANOTARIAN POR MIEDO A NO RECORDARLA Y ELLO INCREMENTARIA LOS RIESGOS DE QUE TRASCIENDA.

AUDITORIA Y CONTROLES DE ACCESO

- v **AUDITORIA.**
- v LA AUDITORIA SUELE REALIZARSE A *POSTERIORI* EN SISTEMAS MANUALES:
 - u SE EXAMINAN LAS RECIENTES TRANSACCIONES DE UNA ORGANIZACION PARA DETERMINAR SI HUBO ILICITOS.
- v LA AUDITORIA EN UN SISTEMA INFORMATICO PUEDE IMPLICAR UN PROCESAMIENTO *INMEDIATO*:
 - u SE VERIFICAN LAS TRANSACCIONES QUE SE ACABAN DE PRODUCIR.
- v UN *REGISTRO DE AUDITORIA* ES UN REGISTRO PERMANENTE DE ACONTECIMIENTOS IMPORTANTES ACAECIDOS EN EL SISTEMA INFORMATICO:
 - u SE REALIZA AUTOMATICAMENTE CADA VEZ QUE OCURRE TAL EVENTO.
 - u SE ALMACENA EN UN AREA ALTAMENTE PROTEGIDA DEL SISTEMA.
 - u ES UN MECANISMO IMPORTANTE DE DETECCION.
- v EL REGISTRO DE AUDITORIA DEBE SER REVISADO CUIDADOSAMENTE Y CON FRECUENCIA:
 - u LAS REVISIONES DEBEN HACERSE:
 - Φ PERIODICAMENTE:
 - SE PRESTA ATENCION REGULARMENTE A LOS PROBLEMAS DE SEGURIDAD.
 - Φ AL AZAR:
 - SE INTENTA ATRAPAR A LOS INTRUSOS DESPREVENIDOS.

AUDITORIA Y CONTROLES DE ACCESO

- v **CONTROLES DE ACCESO.**
- v LO FUNDAMENTAL PARA LA SEGURIDAD INTERNA ES CONTROLAR EL ACCESO A LOS DATOS ALMACENADOS.
- v LOS *DERECHOS DE ACCESO* DEFINEN QUE ACCESO TIENEN VARIOS *SUJETOS* O VARIOS *OBJETOS*.
- v LOS *SUJETOS* ACCEDEN A LOS OBJETOS.
- v LOS *OBJETOS* SON ENTIDADES QUE CONTIENEN INFORMACION.
- v LOS OBJETOS PUEDEN SER:
 - u *CONCRETOS*:
 - Φ EJ.: DISCOS, CINTAS, PROCESADORES, ALMACENAMIENTO, ETC.
 - u *ABSTRACTOS*:
 - Φ EJ.: ESTRUCTURAS DE DATOS, DE PROCESOS, ETC.
- v LOS OBJETOS ESTAN PROTEGIDOS CONTRA LOS SUJETOS.
- v LAS AUTORIZACIONES A UN SISTEMA SE CONCEDEN A LOS SUJETOS.
- v LOS SUJETOS PUEDEN SER VARIOS TIPOS DE ENTIDADES:
 - u EJ.: USUARIOS, PROCESOS, PROGRAMAS, OTRAS ENTIDADES, ETC.
- v LOS *DERECHOS DE ACCESO* MAS COMUNES SON:
 - u *ACCESO DE LECTURA*.
 - u *ACCESO DE ESCRITURA*.
 - u *ACCESO DE EJECUCION*.
- v UNA FORMA DE IMPLEMENTACION ES MEDIANTE UNA *MATRIZ DE CONTROL DE ACCESO* CON:
 - u FILAS PARA LOS SUJETOS.
 - u COLUMNAS PARA LOS OBJETOS.
 - u CELDAS DE LA MATRIZ PARA LOS DERECHOS DE ACCESO QUE UN USUARIO TIENE A UN OBJETO.
- v UNA MATRIZ DE CONTROL DE ACCESO DEBE SER MUY CELOSAMENTE PROTEGIDA POR EL S. O.

NUCLEOS DE SEGURIDAD Y SEGURIDAD POR HARDWARE

- v **NUCLEOS DE SEGURIDAD.**
- v ES MUCHO MAS FACIL HACER UN SISTEMA MAS SEGURO SI LA SEGURIDAD SE HA INCORPORADO DESDE EL PRINCIPIO AL DISEÑO DEL SISTEMA.
- v LAS MEDIDAS DE SEGURIDAD DEBEN SER IMPLEMENTADAS EN TODO EL SISTEMA INFORMATICO.
- v UN SISTEMA DE ALTA SEGURIDAD REQUIERE QUE EL NUCLEO DEL S. O. SEA SEGURO.
- v LAS MEDIDAS DE SEGURIDAD MAS DECISIVAS SE IMPLEMENTAN EN EL NUCLEO:
 - u SE MANTIENE INTENCIONALMENTE LO MAS PEQUEÑO POSIBLE.
- v GENERALMENTE SE DA QUE:
 - u AISLANDO LAS FUNCIONES QUE DEBEN SER ASEGURADAS EN UN S. O. DE PROPOSITO GENERAL A GRAN ESCALA:
 - Φ SE CREA UN NUCLEO GRANDE.
- v LA SEGURIDAD DEL SISTEMA DEPENDE ESPECIALMENTE DE ASEGURAR LAS FUNCIONES QUE REALIZAN:
 - u EL CONTROL DE ACCESO.
 - u LA ENTRADA AL SISTEMA.
 - u LA VERIFICACION.
 - u LA ADMINISTRACION DEL ALMACENAMIENTO REAL, DEL ALMACENAMIENTO VIRTUAL Y DEL SISTEMA DE ARCHIVOS.
- v **SEGURIDAD POR HARDWARE.**
- v EXISTE UNA TENDENCIA A INCORPORAR AL HARDWARE FUNCIONES DEL S. O.:
 - u LAS FUNCIONES INCORPORADAS AL HARDWARE:
 - Φ RESULTAN MUCHO MAS SEGURAS QUE CUANDO SON ASEQUIBLES COMO INSTRUCCIONES DE SOFTWARE QUE PUEDEN SER MODIFICADAS.
 - Φ PUEDEN OPERAR MUCHO MAS RAPIDO QUE EN EL SOFTWARE:
 - MEJORANDO LA PERFORMANCE.
 - PERMITIENDO CONTROLES MAS FRECUENTES.

SISTEMAS SUPERVIVIENTES

- v EL DISEÑO DE SISTEMAS DE ALTA SEGURIDAD DEBE ASEGURAR:
 - u SU OPERACION DE MANERA CONTINUA Y CONFIABLE.
 - u SU DISPONIBILIDAD.
- v UN *SISTEMA DE COMPUTACION SUPERVIVIENTE* ES AQUEL QUE CONTINUA OPERANDO AUN DESPUES DE QUE UNO O MAS DE SUS COMPONENTES FALLA:
 - u ES UNA CUESTION CADA VEZ MAS IMPORTANTE, ESPECIALMENTE PARA SISTEMAS EN LINEA.
- v GENERALMENTE CONTINUAN OPERANDO CON UNA *DEGRADACION SUAVE* EN LOS NIVELES DE PRESTACION.
- v LOS COMPONENTES FALLIDOS DEBEN PODER REEMPLAZARSE SIN INTERRUPTIR EL FUNCIONAMIENTO DEL SISTEMA.
- v UNA CLAVE PARA LA CAPACIDAD DE SUPERVIVENCIA ES LA *REDUNDANCIA*:
 - u SI UN COMPONENTE FALLA, OTRO EQUIVALENTE TOMA SU PUESTO.
 - u SE PUEDE IMPLEMENTAR COMO:
 - Φ UN CONJUNTO DE RECURSOS IDENTICOS QUE FUNCIONAN EN PARALELO.
 - Φ UN CONJUNTO SEPARADO DE RECURSOS REDUNDANTES QUE SE ACTIVAN CUANDO SE PRODUCE UN FALLO.
- v ALGUNAS CARACTERISTICAS DE SUPERVIVENCIA SON:
 - u LA INCORPORACION DE MECANISMOS CONTRA FALLOS EN EL HARDWARE EN VEZ DE EN EL SOFTWARE.
 - u EL USO DE *MULTIPROCESAMIENTO TRANSPARENTE* PARA PERMITIR MEJORAR EL RENDIMIENTO SIN MODIFICAR EL SOFTWARE.
 - u EL USO DE SUBSISTEMAS MULTIPLES DE ENTRADA / SALIDA.
 - u LA INCORPORACION DE MECANISMOS DE DETECCION DE FALLOS EN EL HARDWARE Y EN EL SOFTWARE.

CAPACIDADES Y SISTEMAS ORIENTADOS HACIA EL OBJETO

- v UN *DERECHO DE ACCESO* PERMITE A ALGUN *SUJETO* ACCEDER A ALGUN *OBJETO* DE UNA MANERA PREESTABLECIDA.
- v LOS SUJETOS SON LOS USUARIOS DE LOS SISTEMAS DE COMPUTACION O ENTIDADES QUE ACTUAN EN NOMBRE:
 - DE LOS USUARIOS.
 - DEL SISTEMA.
 - EJ.: TRABAJOS, PROCESOS Y PROCEDIMIENTOS, ETC.
- v LOS OBJETOS SON LOS RECURSOS DEL SISTEMA:
 - EJ.: ARCHIVOS, PROGRAMAS, SEMAFOROS, DIRECTORIOS, TERMINALES, CANALES, DISPOSITIVOS, PISTAS DE DISCOS, BLOQUES DE ALMACENAMIENTO PRIMARIO, ETC.
- v LOS SUJETOS SE CONSIDERAN TAMBIEN COMO OBJETOS DEL SISTEMA:
 - UN SUJETO PUEDE TENER DERECHOS DE ACCEDER A OTRO.
- v LOS SUJETOS SON ENTIDADES ACTIVAS Y LOS OBJETOS SON PASIVOS.
- v UNA *CAPACIDAD* ES UNA SEÑAL:
 - LA POSESION DE UNA CAPACIDAD POR UN SUJETO LE CONFIERE DERECHOS DE ACCESO A UN OBJETO.
- v LAS CAPACIDADES NO SUELEN SER MODIFICADAS PERO SUELEN SER REPRODUCIDAS.
- v UN *DOMINIO DE PROTECCION* DEFINE LOS DERECHOS DE ACCESO QUE UN SUJETO TIENE A LOS DISTINTOS OBJETOS DEL SISTEMA:
 - ES EL CONJUNTO DE CAPACIDADES QUE PERTENECEN AL SUJETO.
- v UNA CAPACIDAD ES UN NOMBRE PROTEGIDO PARA UN OBJETO DEL SISTEMA:
 - EL NOMBRE ES UNICO EN TODO EL SISTEMA.
 - PARA TENER ACCESO A UN OBJETO DETERMINADO, UN SUJETO DEBE POSEER UNA CAPACIDAD PARA HACERLO.
- v LA CAPACIDAD INCLUYE UNA INSTRUCCION DE LOS DERECHOS DE ACCESO DETERMINADOS QUE LA CAPACIDAD LE PERMITE AL SUJETO RESPECTO DEL OBJETO CORRESPONDIENTE.
- v LA CREACION DE CAPACIDADES ES UNA FUNCION DE RUTINAS DE LOS S. O. CUIDADOSAMENTE GUARDADAS.
- v LO NORMAL ES QUE LAS CAPACIDADES NO PUEDEN SER MODIFICADAS SALVO PARA REDUCIR LOS DERECHOS DE ACCESO ESTABLECIDOS.
- v UN SUJETO CON UNA CAPACIDAD PUEDE COPIARLA O PASARLA COMO UN PARAMETRO.

CAPACIDADES Y SISTEMAS ORIENTADOS HACIA EL OBJETO

- v LUEGO DE LA CREACION DE UN OBJETO SE CREA UNA CAPACIDAD PARA ESE OBJETO:
 - u INCLUYE TODOS LOS DERECHOS DE ACCESO AL NUEVO OBJETO.
- v EL SUJETO QUE CREA LA CAPACIDAD PUEDE PASAR COPIAS DE LA CAPACIDAD A OTROS SUJETOS:
 - u LA PUEDEN USAR O COPIARLA A OTROS SUJETOS:
 - Φ SIN VARIANTES.
 - Φ REDUCIENDO (NUNCA INCREMENTANDO) LOS DERECHOS DE ACCESO ESTABLECIDOS.
- v SI SE HAN INTEGRADO CAPACIDADES EN EL HARDWARE DE DIRECCIONAMIENTO DEL ALMACENAMIENTO PRIMARIO:
 - u SE LAS UTILIZA EN CADA REFERENCIA AL MISMO.
 - u TENEMOS UN *DIRECCIONAMIENTO BASADO EN LA CAPACIDAD*.
- v EN SISTEMAS BASADOS EN CAPACIDADES SE PUEDE PRESENTAR EL *PROBLEMA DEL OBJETO PERDIDO*:
 - u SI SE DESTRUYE LA ULTIMA CAPACIDAD RESTANTE DE UN OBJETO, ESTE NO PODRA SER USADO DE NINGUNA MANERA.
 - u EL SISTEMA DEBE MANTENER SIEMPRE AL MENOS UNA CAPACIDAD PARA CADA OBJETO.
- v EL CONTROL DEL COPIADO Y MOVIMIENTO DE LAS CAPACIDADES ES UN PROBLEMA DIFICIL:
 - u GENERALMENTE EL S. O. REALIZA LA MANIPULACION DE CAPACIDADES EN NOMBRE DE LOS USUARIOS.
- v LA REVOCACION DE LAS CAPACIDADES PASADAS A OTRO SUJETO TAMBIEN PUEDE COMPLICARSE:
 - u LA CAPACIDAD PUDO HABER SIDO COPIADA MUCHAS VECES.
 - u PODRIA SER NECESARIO REVOCAR LA CAPACIDAD:
 - Φ DE UN SUJETO DETERMINADO.
 - Φ LA DE CUALQUIER OTRO SUJETO QUE HUBIERA RECIBIDO DE EL ESA CAPACIDAD.
- v UNA TECNICA PARA LA *REVOCACION SELECTIVA DE LAS CAPACIDADES* ES LA SIGUIENTE:
 - u TODAS LAS CAPACIDADES CREADAS A PARTIR DE UNA PRINCIPAL, APUNTEN AL OBJETO A TRAVES DE LA CAPACIDAD PRINCIPAL.

CRIPTOGRAFIA

- v EL USO CRECIENTE DE LAS REDES DE COMPUTADORAS Y LA IMPORTANCIA DEL TRAFICO CURSADO HACE NECESARIO PROTEGER A LOS DATOS.
- v LA OFICINA NACIONAL DE ESTANDARES DE EE. UU. (NBS) HA ADOPTADO LA NORMA DE CIFRADO DE DATOS (DES) PARA LA TRANSMISION DE INFORMACION FEDERAL DELICADA.
- v LA *CRIPTOGRAFIA* ES EL USO DE LA TRANSFORMACION DE DATOS PARA HACERLOS INCOMPENSIBLES A TODOS:
 - u EXCEPTO A LOS USUARIOS A QUIENES ESTAN DESTINADOS.
- v EL *PROBLEMA DE LA INTIMIDAD* TRATA DE COMO EVITAR LA OBTENCION NO AUTORIZADA DE INFORMACION DE UN CANAL DE COMUNICACIONES.
- v EL *PROBLEMA DE LA AUTENTIFICACION* TRATA SOBRE COMO EVITAR QUE UN Oponente:
 - u MODIFIQUE UNA TRANSMISION.
 - u LE INTRODUZCA DATOS FALSOS.
- v EL *PROBLEMA DE LA DISPUTA* TRATA SOBRE COMO PROPORCIONAR AL RECEPTOR DE UN MENSAJE PRUEBAS LEGALES DE LA IDENTIDAD DEL REMITENTE:
 - u EL EQUIVALENTE ELECTRONICO DE UNA FIRMA ESCRITA.
- v **UN SISTEMA DE INTIMIDAD CRIPTOGRAFICA.**
- v EL *REMITENTE* DESEA TRANSMITIR CIERTO MENSAJE NO CIFRADO (*TEXTO SIMPLE*) A UN *RECEPTOR* LEGITIMO:
 - u LA TRANSMISION SE PRODUCIRA A TRAVES DE UN *CANAL INSEGURO*:
 - Φ SE SUPONE QUE PODRA SER VERIFICADO O CONECTADO MEDIANTE UN *ESPIA*.
- v EL REMITENTE PASA EL TEXTO SIMPLE A UNA UNIDAD DE CODIFICACION QUE LO TRANSFORMA EN UN *TEXTO CIFRADO* O *CRIPTOGRAMA*:
 - u NO ES COMPENSIBLE PARA EL *ESPIA*.
 - u SE TRANSMITE EN FORMA SEGURA POR UN CANAL INSEGURO.
 - u EL RECEPTOR PASA EL TEXTO CIFRADO POR UNA UNIDAD DE DESCIFRADO PARA REGENERAR EL TEXTO SIMPLE.

CRIPTOGRAFIA

- v **CRIPTOANALISIS.**

- v ES EL PROCESO DE INTENTAR REGENERAR EL TEXTO SIMPLE A PARTIR DEL TEXTO CIFRADO, PERO DESCONOCIENDO LA CLAVE DE CIFRAMIENTO:

- v ES LA TAREA DEL ESPIA O *CRIPTOANALISTA*:
 - Φ SI NO LO LOGRA, EL SISTEMA CRIPTOGRAFICO ES *SEGURO*.

- v **SISTEMAS DE CLAVE PUBLICA.**

- v LA DISTRIBUCION DE CLAVES DE UN SISTEMA CRIPTOGRAFICO DEBE HACERSE POR CANALES MUY SEGUROS.

- v LOS *SISTEMAS DE CLAVE PUBLICA* RODEAN EL PROBLEMA DE DISTRIBUCION DE CLAVES:

- v LAS FUNCIONES DE CIFRADO Y DESCIFRADO ESTAN SEPARADAS Y UTILIZAN DISTINTAS CLAVES.
 - v NO ES COMPUTACIONALMENTE POSIBLE (EN UN TIEMPO “RAZONABLE”) DETERMINAR LA CLAVE DE DESCIFRAMIENTO “D” A PARTIR DE LA CLAVE DE CIFRAMIENTO “C”.
 - v “C” PUEDE HACERSE *PUBLICA* SIN COMPROMETER LA SEGURIDAD DE “D”, QUE PERMANECE *PRIVADA*:
 - Φ SE SIMPLIFICA EL PROBLEMA DE LA DISTRIBUCION DE CLAVES.

- v **FIRMAS DIGITALES.**

- v PARA QUE UNA *FIRMA DIGITAL* SEA ACEPTADA COMO SUSTITUTA DE UNA FIRMA ESCRITA DEBE SER:

- v FACIL DE AUTENTICAR (RECONOCER) POR CUALQUIERA.
 - v PRODUCIBLE UNICAMENTE POR SU AUTOR.

- v EN LOS CRIPTOSISTEMAS DE CLAVE PUBLICA EL PROCEDIMIENTO ES:

- v EL REMITENTE USA LA CLAVE PRIVADA PARA CREAR UN MENSAJE FIRMADO.
 - v EL RECEPTOR:
 - Φ USA LA CLAVE PUBLICA DEL REMITENTE PARA DESCIFRAR EL MENSAJE.
 - Φ GUARDA EL MENSAJE FIRMADO PARA USARLO EN CASO DE DISPUTAS.

CRİPTOGRAFIA

- v PARA MAYOR SEGURIDAD SE PODRIA ACTUAR COMO SIGUE:
 - u EL REMITENTE PUEDE CODIFICAR EL MENSAJE YA CIFRADO UTILIZANDO LA CLAVE PUBLICA DEL RECEPTOR.
 - u LA CLAVE PRIVADA DEL RECEPTOR PERMITE RECUPERAR EL MENSAJE CIFRADO FIRMADO.
 - u LA CLAVE PUBLICA DEL REMITENTE PERMITE RECUPERAR EL TEXTO SIMPLE ORIGINAL.
- v **APLICACIONES.**
- v LA CRİPTOGRAFIA ES ESPECIALMENTE UTIL EN LOS SISTEMAS MULTIUSUARIO Y EN LAS REDES DE COMPUTADORAS.
- v SE DEBE UTILIZAR PARA PROTEGER A LAS CONTRASEÑAS, ALMACENANDOLAS CIFRADAS.
- v SE PUEDE UTILIZAR TAMBIEN PARA PROTEGER TODOS LOS DATOS ALMACENADOS EN UN SISTEMA DE COMPUTACION:
 - u SE DEBE CONSIDERAR EL TIEMPO DE CIFRADO / DESCIFRADO.
- v TAMBIEN ES APLICABLE EN LOS PROTOCOLOS DE REDES DE CAPAS:
 - u OFRECEN VARIOS NIVELES DE CIFRADO.
- v EN EL *CIFRADO DE ENLACE* LA RED ASUME LA RESPONSABILIDAD DE CIFRADO / DESCIFRADO DE CADA NODO:
 - u LOS DATOS SE TRANSMITEN CIFRADOS ENTRE LOS NODOS.
 - u EN CADA NODO SE DESCIFRAN, SE DETERMINA A DONDE TRANSMITIRLOS Y SE LOS VUELVE A CIFRAR.
- v EN EL *CIFRADO PUNTO A PUNTO* UN MENSAJE SE CIFRA EN SU FUENTES Y SE DESCIFRA SOLO UNA VEZ, EN SU DESTINO:
 - u EXISTEN CIERTAS LIMITACIONES TALES COMO LA LEGIBILIDAD DE LA DIRECCION DE DESTINO EN CADA NODO:
 - ⦿ DEBE SER LEGIBLE PARA EL ENCAMINAMIENTO DEL MENSAJE.
 - ⦿ EJ.: SISTEMAS DE CONMUTACION DE PAQUETES DE ALMACENAMIENTO Y REENVIO CON CIFRADO PUNTO A PUNTO:
 - LA DIRECCION DE DESTINO ASOCIADA A UN PAQUETE NO PUEDE SER CIFRADA.

PENETRACION AL SISTEMA OPERATIVO

- v LA PENETRACION DEFINITIVA PUEDE CONSISTIR EN CAMBIAR EL BIT DE ESTADO DE LA MAQUINA DEL ESTADO PROBLEMA AL ESTADO SUPERVISOR:
 - v EL INTRUSO PODRA ASI EJECUTAR INSTRUCCIONES PRIVILEGIADAS PARA OBTENER ACCESO A LOS RECURSOS PROTEGIDOS POR EL S. O.
- v LOS *ESTUDIOS DE PENETRACION* ESTAN DISEÑADOS PARA:
 - v DETERMINAR SI LAS DEFENSAS DE UN SISTEMA CONTRA ATAQUES DE USUARIOS NO PRIVILEGIADOS SON ADECUADAS.
 - v DESCUBRIR DEFICIENCIAS DE DISEÑO PARA CORREGIRLAS.
- v EL CONTROL DE ENTRADA / SALIDA ES UN AREA FAVORITA PARA INTENTAR LA PENETRACION A UN SISTEMA:
 - v LOS CANALES DE ENTRADA / SALIDA TIENEN ACCESO AL ALMACENAMIENTO PRIMARIO:
 - ❖ PUEDEN MODIFICAR INFORMACION IMPORTANTE.
- v UNA DE LAS METAS DE LAS PRUEBAS DE PENETRACION CONSISTE EN ESTIMAR EL *FACTOR DE TRABAJO DE PENETRACION*:
 - v INDICACION DE CUANTO ESFUERZO Y RECURSOS SON NECESARIOS PARA CONSEGUIR UN ACCESO NO AUTORIZADO A LOS RECURSOS DEL SISTEMA:
 - ❖ DEBERIA SER TAN GRANDE QUE RESULTE DISUASIVO.
- v **PRINCIPALES FALLOS GENERICOS FUNCIONALES DE LOS SISTEMAS.**
- v *AUTENTIFICACION*:
 - v LOS USUARIOS NO PUEDEN DETERMINAR SI EL HARDWARE Y EL SOFTWARE CON QUE FUNCIONAN SON LOS QUE DEBEN SER.
 - v UN INTRUSO PODRIA REEMPLAZAR UN PROGRAMA SIN CONOCIMIENTO DEL USUARIO.
 - v UN USUARIO PUEDE INADVERTIDAMENTE TECLEAR UNA CONTRASEÑA EN UN PROGRAMA DE ENTRADA FALSO.
- v *CIFRADO*:
 - v NO SE ALMACENA CIFRADA LA LISTA MAESTRA DE CONTRASEÑAS.

PENETRACION AL SISTEMA OPERATIVO

- v *IMPLEMENTACION:*
 - υ IMPLEMENTACION IMPROCEDENTE DE UN BUEN DISEÑO DE SEGURIDAD.
- v *CONFIANZA IMPLICITA:*
 - υ UNA RUTINA SUPONE QUE OTRA ESTA FUNCIONANDO CORRECTAMENTE CUANDO, DE HECHO, DEBERIA EXAMINAR LOS PARAMETROS SUMINISTRADOS POR LA OTRA RUTINA.
- v *COMPARTIMIENTO IMPLICITO:*
 - υ EL S. O. DEPOSITA INADVERTIDAMENTE INFORMACION IMPORTANTE DEL SISTEMA EN UN ESPACIO DE DIRECCIONES DEL USUARIO.
- v *COMUNICACION ENTRE PROCESOS:*
 - υ USOS INADECUADOS DE LOS MECANISMOS DE SEND / RECEIVE QUE PUEDEN SER APROVECHADOS POR LOS INTRUSOS.
- v *VERIFICACION DE LA LEGALIDAD:*
 - υ VALIDACION INSUFICIENTE DE LOS PARAMETROS DEL USUARIO.
- v *DESCONEXION DE LINEA:*
 - υ ANTE UNA DESCONEXION DE LINEA EL S. O. DEBERIA:
 - Φ DAR DE BAJA AL USUARIO (O LOS USUARIOS) DE LA LINEA.
 - Φ COLOCARLOS EN UN ESTADO TAL QUE REQUIERA LA RE - AUTORIZACION PARA OBTENER NUEVAMENTE EL CONTROL.
- v *DESCUIDO DEL OPERADOR:*
 - υ UN INTRUSO PODRIA ENGAÑAR A UN OPERADOR Y HACER QUE LE HABILITE DETERMINADOS RECURSOS.
- v *PASO DE PARAMETROS POR REFERENCIA EN FUNCION DE SU VALOR:*
 - υ ES MAS SEGURO PASAR LOS PARAMETROS DIRECTAMENTE EN REGISTROS QUE TENER LOS REGISTROS APUNTANDO A LAS AREAS QUE CONTIENEN LOS PARAMETROS.
 - υ EL PASO POR REFERENCIA PUEDE PERMITIR QUE LOS PARAMETROS, ESTANDO AUN EN EL AREA DEL USUARIO, PUEDAN SER MODIFICADOS ANTES DE SER USADOS POR EL SISTEMA.
- v *CONTRASEÑAS:*
 - υ NO DEBEN SER FACILMENTE DEDUCIBLES U OBTENIBLES MEDIANTE ENSAYOS REPETIDOS.

PENETRACION AL SISTEMA OPERATIVO

- v *ENTRAMPAMIENTO AL INTRUSO:*
 - u LOS S. O. DEBEN TENER MECANISMOS DE ENTRAMPAMIENTO PARA ATRAER AL INTRUSO INEXPERTO.
- v *PRIVILEGIO:*
 - u CUANDO HAY DEMASIADOS PROGRAMAS CON DEMASIADOS PRIVILEGIOS SE VIOLA EL *PRINCIPIO DEL MENOR PRIVILEGIO*.
- v *CONFINAMIENTO DEL PROGRAMA:*
 - u UN PROGRAMA “PRESTADO” DE OTRO USUARIO PUEDE ACTUAR COMO UN “CABALLO DE TROYA”:
- v *PROHIBICIONES:*
 - u SE ADVIERTE A LOS USUARIOS QUE NO UTILICEN CIERTAS OPCIONES PORQUE LOS RESULTADOS PODRIAN SER “INDETERMINADOS”, PERO NO SE BLOQUEA SU USO.
 - Φ PUEDE ROBAR O ALTERAR DATOS.
- v *RESIDUOS:*
 - u UN INTRUSO PODRIA ENCONTRAR UNA LISTA DE CONTRASEÑAS CON SOLO BUSCAR EN LUGARES TALES COMO UNA “PAPELERA”:
 - Φ DEL SISTEMA O FISICA.
 - Φ LA INFORMACION DELICADA DEBE SER SOBRESCRITA O DESTRUIDA ANTES DE LIBERAR O DESCARTAR EL MEDIO QUE OCUPA.
- v *BLINDAJE:*
 - u LOS INTRUSOS PUEDEN CONECTARSE A UNA LINEA DE TRANSMISION SIN HACER CONTACTO FISICO:
 - Φ UTILIZAN EL CAMPO INDUCIDO POR LA CIRCULACION DE CORRIENTE EN UN CABLE.
 - Φ SE PREVIENE CON UN ADECUADO BLINDAJE ELECTRICO.
- v *VALORES DE UMBRAL:*
 - u SI NO SE DISPONE DE VALORES UMBRAL, NO HABRA:
 - Φ LIMITES AL N° DE INTENTOS FALLIDOS DE INGRESO.
 - Φ BLOQUEOS A NUEVOS INTENTOS.
 - Φ COMUNICACIONES AL SUPERVISOR O ADMINISTRADOR DEL SISTEMA.

PENETRACION AL SISTEMA OPERATIVO

v **ATAQUES GENERICOS A SISTEMAS OPERATIVOS.**

v **ASINCRONISMO:**

- v SE TIENEN PROCESOS MULTIPLES QUE PROGRESAN ASINCRONICAMENTE.
- v UN PROCESO PODRIA MODIFICAR LOS PARAMETROS YA VALIDADOS POR OTRO PROCESO PERO AUN NO UTILIZADOS.
- v UN PROCESO PODRIA PASAR VALORES MALOS A OTRO AUN CUANDO EL SEGUNDO REALICE UNA VERIFICACION EXTENSA.

v **RASTREO:**

- v UN USUARIO REVISA EL SISTEMA INTENTANDO LOCALIZAR INFORMACION PRIVILEGIADA.

v **ENTRE LINEAS:**

- v SE UTILIZA UNA LINEA DE COMUNICACIONES MANTENIDA POR UN USUARIO HABILITADO QUE ESTA INACTIVO.

v **CODIGO CLANDESTINO:**

- v SE MODIFICA EL S. O. BAJO UNA PRESUNTA DEPURACION PERO SE INCORPORA CODIGO QUE PERMITE INGRESOS NO AUTORIZADOS.

v **PROHIBICION DE ACCESO:**

- v UN USUARIO ESCRIBE UN PROGRAMA QUE BLOQUEA EL ACCESO O SERVICIO A LOS USUARIOS LEGITIMOS MEDIANTE:
 - Φ CAIDAS DEL SISTEMA, CICLOS INFINITOS, MONOPOLIO DE RECURSOS, ETC.

v **PROCESOS SINCRONIZADOS INTERACTIVOS:**

- v SE UTILIZAN LAS PRIMITIVAS DE SINCRONIZACION DEL SISTEMA PARA COMPARTIR Y PASARSE INFORMACION ENTRE SI.

v **DESCONEXION DE LINEA:**

- v EL INTRUSO INTENTA ACCEDER AL TRABAJO DE UN USUARIO DESCONECTADO:
 - Φ LUEGO DE UNA DESCONEXION DE LINEA.
 - Φ ANTES DE QUE EL SISTEMA RECONOZCA LA DESCONEXION.

PENETRACION AL SISTEMA OPERATIVO

v *DISFRAZ:*

- υ EL INTRUSO ASUME LA IDENTIDAD DE UN USUARIO LEGITIMO LUEGO DE HABER OBTENIDO LA IDENTIFICACION APROPIADA POR MEDIOS CLANDESTINOS.

v *ATAQUE "NAK":*

- υ SI EL S. O. PERMITE A UN USUARIO:
 - Φ INTERRUMPIR UN PROCESO EN EJECUCION MEDIANTE UNA "TECLA" DE "RECONOCIMIENTO NEGATIVO".
 - Φ REALIZAR OTRA OPERACION.
 - Φ REANUDAR EL PROCESO INTERRUMPIDO.
- υ UN INTRUSO PODRIA "ENCONTRAR" AL SISTEMA EN UN ESTADO NO PROTEGIDO Y HACERSE CON EL CONTROL.

v *ENGAÑO AL OPERADOR:*

- υ CON UN ENGAÑO SE HACE REALIZAR AL OPERADOR UNA ACCION QUE COMPROMETA LA SEGURIDAD DEL SISTEMA.

v *PARASITO:*

- υ MEDIANTE EQUIPAMIENTO ESPECIAL EL INTRUSO:
 - Φ INTERCEPTA LOS MENSAJES ENTRE UN USUARIO HABILITADO Y EL PROCESADOR.
 - Φ LOS MODIFICA O REEMPLAZA TOTALMENTE.

v *CABALLO DE TROYA:*

- υ EL INTRUSO COLOCA UN CODIGO DENTRO DEL SISTEMA QUE LUEGO LE PERMITA ACCESOS NO AUTORIZADOS.
- υ PUEDE PERMANECER EN EL SISTEMA.
- υ PUEDE BORRAR TODO RASTRO DE SI MISMO LUEGO DE LA PENETRACION.

v *PARAMETROS INESPERADOS:*

- υ EL INTRUSO SUMINISTRA VALORES INESPERADOS A UNA LLAMADA AL NUCLEO.
- υ INTENTA APROVECHAR UNA DEBILIDAD DE LOS MECANISMOS DE VERIFICACION DE LA LEGALIDAD DEL S. O.

ESTUDIO Y COMPARACION DE DISTINTOS SISTEMAS OPERATIVOS

- TRABAJOS TEORICO - PRACTICOS CON PRESENTACION DE MONOGRAFIAS

INDICE GENERAL

■	INTRODUCCION	003
◆	QUE ES UN SISTEMA OPERATIVO	004
◆	HISTORIA DE LOS SISTEMAS OPERATIVOS - GENERACIONES	008
◆	CONCEPTOS DE LOS SISTEMAS OPERATIVOS	011
◆	ESTRUCTURA DE LOS SISTEMAS OPERATIVOS	013
◆	TENDENCIAS	021
◆	HARDWARE	022
◆	SOFTWARE	028
◆	MEMORIA FIJA	032
■	PROCESOS Y ADMINISTRACION DEL PROCESADOR	035
◆	INTRODUCCION Y DEFINICIONES SOBRE PROCESOS	036
◆	ESTADOS DE PROCESO	039
◆	PROCESAMIENTO DE INTERRUPCIONES	041
◆	EL NUCLEO DEL S. O.	044
◆	PLANIFICACION DE PROCESOS	045
◆	NIVELES DE PLANIFICACION DEL PROCESADOR	047
◆	OBJETIVOS DE LA PLANIFICACION	049
◆	CRITERIOS DE PLANIFICACION	051
◆	PLANIFICACION APROPIATIVA VERSUS NO APROPIATIVA	053
◆	TEMPORIZADOR DE INTERVALOS O RELOJ DE INTERRUPCIONES	054
◆	PRIORIDADES	055
◆	TIPOS DE PLANIFICACION	056
◆	MULTIPROCESAMIENTO	066
◆	ORGANIZACION DEL HARDWARE DEL MULTIPROCESADOR	072
◆	GRADOS DE ACOPLAMIENTO EN MULTIPROCESAMIENTO	076

INDICE GENERAL

◆ S. O. DE MULTIPROCESADORES	078
◆ RENDIMIENTO DEL SISTEMA DE MULTIPROCESAMIENTO	080
◆ RECUPERACION DE ERRORES	080
◆ MULTIPROCESAMIENTO SIMETRICO	081
◆ TENDENCIAS DE LOS MULTIPROCESADORES	083
■ ADMINISTRACION DE LA MEMORIA	084
◆ ALMACENAMIENTO REAL:	085
◆ INTRODUCCION	085
◆ ORGANIZACION Y ADMINISTRACION DEL ALMACENAMIENTO	086
◆ JERARQUIA DE ALMACENAMIENTO	087
◆ ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO	089
◆ MULTIPROGRAMACION DE PARTICION FIJA	093
◆ MULTIPROGRAMACION DE PARTICION VARIABLE	097
◆ MULTIPROGRAMACION CON INTERCAMBIO DE ALMACENAMIENTO	101
◆ ORGANIZACION DEL ALMACENAMIENTO VIRTUAL:	102
◆ INTRODUCCION	102
◆ CONCEPTOS BASICOS DE ALMACENAMIENTO VIRTUAL	103
◆ ORGANIZACION DEL ALMACENAMIENTO DE NIVELES MULTIPLES	105
◆ TRANSFORMACION DE BLOQUES	107
◆ CONCEPTOS BASICOS DE PAGINACION	110
◆ SEGMENTACION	120
◆ SISTEMAS DE PAGINACION / SEGMENTACION	127

INDICE GENERAL

◆	ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL:	133
◆	INTRODUCCION	133
◆	ESTRATEGIAS DE ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL	133
◆	LOCALIDAD	137
◆	CONJUNTOS DE TRABAJO	140
◆	PAGINACION POR DEMANDA Y PAGINACION ANTICIPADA	142
◆	LIBERACION DE PAGINA Y TAMAÑO DE PAGINA	143
◆	COMPORTAMIENTO DE UN PROGRAMA EN LA PAGINACION	145
■	SISTEMAS DE ARCHIVOS	147
◆	INTRODUCCION	148
◆	FUNCIONES DEL SISTEMA DE ARCHIVOS	149
◆	EL SISTEMA DE ARCHIVOS	150
◆	ARCHIVOS	151
◆	DIRECTORIOS	156
◆	IMPLANTACION DEL SISTEMA DE ARCHIVOS Y SU RELACION CON LA ASIGNACION Y LIBERACION DE ESPACIO	161
◆	DESCRIPTOR DE ARCHIVOS	183
◆	SEGURIDAD	184
◆	MECANISMOS DE PROTECCION	190
◆	RESPALDO Y RECUPERACION	195
■	ENTRADA / SALIDA	196
◆	INTRODUCCION	197
◆	PRINCIPIOS DEL HARDWARE DE E / S	198
◆	PRINCIPIOS DEL SOFTWARE DE E / S	204

INDICE GENERAL

◆ DISCOS	210
◆ HARDWARE PARA DISCOS	210
◆ OPERACION DE ALMACENAMIENTO DE DISCO DE CABEZA MOVIL	211
◆ ALGORITMOS DE PROGRAMACION DEL BRAZO DEL DISCO	213
◆ PORQUE ES NECESARIA LA PLANIFICACION DE DISCOS	217
◆ CARACTERISTICAS DESEABLES DE LAS POLITICAS DE PLANIFICACION DE DISCOS	218
◆ OPTIMIZACION DE LA BUSQUEDA EN DISCOS	219
◆ OPTIMIZACION ROTACIONAL EN DISCOS	221
◆ CONSIDERACIONES DE LOS DISCOS SOBRE LOS SISTEMAS	221
◆ MANEJO DE ERRORES EN DISCOS	223
◆ OCULTAMIENTO DE UNA PISTA A LA VEZ EN DISCOS	224
◆ DISCOS EN RAM	225
◆ RELOJES	225
◆ TERMINALES	226
■ BLOQUEOS	229
◆ INTRODUCCION Y EJEMPLOS DE BLOQUEO (O INTERBLOQUEO)	230
◆ CONCEPTOS DE RECURSOS	233
◆ BLOQUEOS Y CONDICIONES NECESARIAS PARA EL BLOQUEO	234
◆ MODELACION DE BLOQUEOS	235
◆ AREAS PRINCIPALES EN LA INVESTIGACION DE BLOQUEOS	238

INDICE GENERAL

◆	EL ALGORITMO DEL AVESTRUZ O DE OSTRICH	239
◆	DETECCION DE BLOQUEOS	240
◆	RECUPERACION DE BLOQUEOS	249
◆	EVASION DE BLOQUEOS	252
◆	PREVENCION DE BLOQUEOS	259
◆	OTROS ASPECTOS	262
◆	TENDENCIAS DEL TRATAMIENTO DEL BLOQUEO	264
■	INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS	265
◆	INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS	266
◆	VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LOS CENTRALIZADOS	267
◆	VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS CON RESPECTO A LAS PC INDEPENDIENTES	269
◆	DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS	270
◆	CONCEPTOS DE HARDWARE	271
◆	MULTIPROCESADORES CON BASE EN BUSES	273
◆	MULTIPROCESADORES CON CONMUTADOR	275
◆	MULTICOMPUTADORAS CON BASE EN BUSES	279
◆	MULTICOMPUTADORAS CON CONMUTADOR	280
◆	CONCEPTOS DE SOFTWARE	282
◆	SISTEMAS OPERATIVOS DE REDES	283
◆	SISTEMAS REALMENTE DISTRIBUIDOS	293
◆	SISTEMAS DE MULTIPROCESADOR CON TIEMPO COMPARTIDO	294
◆	ASPECTOS DEL DISEÑO	296
◆	TRANSPARENCIA	297
◆	FLEXIBILIDAD	298
◆	CONFIABILIDAD	300
◆	DESEMPEÑO	301
◆	ESCALABILIDAD	302

INDICE GENERAL

■	COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS	303
◆	INTRODUCCION A LA COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS	304
◆	PROTOCOLOS CON CAPAS	305
◆	INTRODUCCION AL MODELO CLIENTE - SERVIDOR (C - S)	308
◆	DIRECCIONAMIENTO EN C - S	310
◆	PRIMITIVAS DE BLOQUEO VS. NO BLOQUEO EN C - S	313
◆	PRIMITIVAS ALMACENADAS EN BUFFER VS. NO ALMACENADAS EN C - S	315
◆	PRIMITIVAS CONFIABLES VS. NO CONFIABLES EN C - S	318
◆	IMPLANTACION DEL MODELO C - S	320
◆	LLAMADA A UN PROCEDIMIENTO REMOTO (RPC)	323
◆	OPERACION BASICA DE RPC	324
◆	TRANSFERENCIA DE PARAMETROS EN RPC	328
◆	CONEXION DINAMICA EN RPC	330
◆	SEMANTICA DE RPC EN PRESENCIA DE FALLOS	332
◆	ASPECTOS DE LA IMPLANTACION DE RPC	338
◆	AREAS DE PROBLEMAS EN RPC	347
◆	COMUNICACION EN GRUPO	349
◆	ASPECTOS DEL DISEÑO DE LA COMUNICACION EN GRUPO	351
■	SINCRONIZACION EN SISTEMAS DISTRIBUIDOS	361
◆	INTRODUCCION A LA SINCRONIZACION EN SISTEMAS DISTRIBUIDOS	362
◆	SINCRONIZACION DE RELOJES	363

INDICE GENERAL

◆ RELOJES LOGICOS	364
◆ RELOJES FISICOS	368
◆ ALGORITMOS PARA LA SINCRONIZACION DE RELOJES	370
◆ EXCLUSION MUTUA	374
◆ ALGORITMOS DE ELECCION	378
◆ TRANSACCIONES ATOMICAS	380
◆ EL MODELO DE TRANSACCION	381
◆ IMPLANTACION DEL MODELO DE TRANSACCION	383
◆ CONTROL DE CONCURRENCIA EN EL MODELO DE TRANSACCION	386
◆ BLOQUEOS EN SISTEMAS DISTRIBUIDOS	390
◆ DETECCION DISTRIBUIDA DE BLOQUEOS	391
◆ PREVENCION DISTRIBUIDA DE BLOQUEOS	393
■ PROCESOS Y PROCESADORES EN SISTEMAS DISTRIBUIDOS	394
◆ INTRODUCCION A LOS HILOS (THREADS)	395
◆ USO DE HILOS	396
◆ ASPECTOS DEL DISEÑO DE UN PAQUETE DE HILOS	397
◆ IMPLANTACION DE UN PAQUETE DE HILOS	398
◆ HILOS Y RPC	400
◆ MODELOS DE SISTEMAS	401
◆ EL MODELO DE ESTACION DE TRABAJO	401
◆ USO DE ESTACIONES DE TRABAJO INACTIVAS	403
◆ EL MODELO DE LA PILA DE PROCESADORES	405
◆ ASIGNACION DE PROCESADORES	406
◆ MODELOS DE ASIGNACION	406
◆ ASPECTOS DEL DISEÑO DE ALGORITMOS DE ASIGNACION DE PROCESADORES	407

INDICE GENERAL

◆	ASPECTOS DE LA IMPLANTACION DE ALGORITMOS DE ASIGNACION DE PROCESADORES	408
◆	EJEMPLOS DE ALGORITMOS DE ASIGNACION DE PROCESADORES	409
◆	PLANIFICACION EN SISTEMAS DISTRIBUIDOS	414
■	SISTEMAS DISTRIBUIDOS DE ARCHIVOS	415
◆	INTRODUCCION A LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS	416
◆	DISEÑO DE LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS	417
◆	LA INTERFAZ DEL SERVICIO DE ARCHIVOS	417
◆	LA INTERFAZ DEL SERVIDOR DE DIRECTORIOS	418
◆	SEMANTICA DE LOS ARCHIVOS COMPARTIDOS	420
◆	IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS	422
◆	USO DE ARCHIVOS	422
◆	ESTRUCTURA DEL SISTEMA	423
◆	OCULTAMIENTO	425
◆	REPLICA	429
◆	CONCLUSIONES IMPORTANTES RESPECTO DE LA IMPLANTACION DE UN SISTEMA DISTRIBUIDO DE ARCHIVOS	431
◆	TENDENCIAS EN LOS SISTEMAS DISTRIBUIDOS DE ARCHIVOS	432
◆	CONSIDERACIONES RESPECTO DEL HARDWARE	432
◆	ESCALABILIDAD	433
◆	REDES EN UN AREA AMPLIA	433
◆	USUARIOS MOVILES	434
◆	TOLERANCIA DE FALLOS	434

INDICE GENERAL

■	RENDIMIENTO	435
◆	INTRODUCCION A LA MEDICION, CONTROL Y EVALUACION DEL RENDIMIENTO	436
◆	TENDENCIAS IMPORTANTES QUE AFECTAN A LOS ASPECTOS DEL RENDIMIENTO	437
◆	NECESIDAD DEL CONTROL Y DE LA EVALUACION DEL RENDIMIENTO	438
◆	MEDICIONES DEL RENDIMIENTO	439
◆	TECNICAS DE EVALUACION DEL RENDIMIENTO	441
◆	EMBOTELLAMIENTOS Y SATURACION	444
◆	CICLOS DE RETROALIMENTACION	445
■	MODELADO ANALITICO EN RELACION AL RENDIMIENTO	446
◆	INTRODUCCION AL MODELADO ANALITICO Y TEORIA DE COLAS	447
◆	FUENTE, LLEGADAS Y LLEGADAS DE POISSON	448
◆	TIEMPOS DE SERVICIO, CAPACIDAD DE LA COLA Y NUMERO DE SERVIDORES EN EL SISTEMA	449
◆	DISCIPLINAS DE COLAS	450
◆	INTENSIDAD DE TRAFICO Y UTILIZACION DEL SERVIDOR	451
◆	ESTADO ESTABLE EN FUNCION DE SOLUCIONES TRANSITORIAS	452
◆	RESULTADO DE LITTLE	452
◆	RESUMEN DEL PROCESO DE POISSON	453
◆	ANALISIS DE UN SISTEMA DE COLAS M / M / 1	456
◆	ANALISIS DE UN SISTEMA DE COLAS M / M / C	458
◆	PROCESOS DE MARKOV	460
◆	PROCESOS DE NACIMIENTO Y MUERTE	461
◆	ANALISIS DEL RENDIMIENTO DE UN SUBSISTEMA DE DISCO	462

INDICE GENERAL

■	SEGURIDAD DE LOS SISTEMAS OPERATIVOS	466
◆	INTRODUCCION A LA SEGURIDAD DE LOS SISTEMAS OPERATIVOS	467
◆	REQUISITOS DE SEGURIDAD	468
◆	UN TRATAMIENTO TOTAL DE LA SEGURIDAD	468
◆	SEGURIDAD EXTERNA Y SEGURIDAD OPERACIONAL	469
◆	VIGILANCIA, VERIFICACION DE AMENAZAS Y AMPLIFICACION	471
◆	PROTECCION POR CONTRASEÑA	472
◆	AUDITORIA Y CONTROLES DE ACCESO	473
◆	NUCLEOS DE SEGURIDAD Y SEGURIDAD POR HARDWARE	475
◆	SISTEMAS SUPERVIVIENTES	476
◆	CAPACIDADES Y SISTEMAS ORIENTADOS HACIA EL OBJETO	477
◆	CRIPTOGRAFIA	479
◆	PENETRACION AL SISTEMA OPERATIVO	482
■	ESTUDIO Y COMPARACION DE DISTINTOS SISTEMAS OPERATIVOS	487
◆	TRABAJOS TEORICO - PRACTICOS CON PRESENTACION DE MONOGRAFIAS	487

INDICE DE MATERIAS

■ A:

- ◆ ALGORITMOS DE ELECCION 378
- ◆ ALMACENAMIENTO 086, 087, 089
- ◆ ALMACENAMIENTO DE DISCO 211
- ◆ ALMACENAMIENTO DE NIVELES MULTIPLES 105
- ◆ ALMACENAMIENTO REAL 085
- ◆ ALMACENAMIENTO VIRTUAL 103, 133
- ◆ AMPLIFICACION 471
- ◆ ARCHIVOS 151, 416, 417, 422
- ◆ ARCHIVOS COMPARTIDOS 420
- ◆ AUDITORIA 473

■ B:

- ◆ BLOQUEO 230, 234, 238
- ◆ BLOQUEOS 240, 249, 252, 259, 390, 391, 393
- ◆ BLOQUES 107
- ◆ BRAZO DEL DISCO 213
- ◆ BUSQUEDA 219

■ C:

- ◆ CAPACIDAD DE LA COLA 449
- ◆ CAPACIDADES 477
- ◆ COLAS 450
- ◆ COLAS M / M / 1 456
- ◆ COLAS M / M / C 458
- ◆ COMUNICACION EN GRUPO 349, 351

INDICE DE MATERIAS

◆	COMUNICACION EN LOS SISTEMAS DISTRIBUIDOS	304
◆	CONCURRENCIA	386
◆	CONFIABILIDAD	300
◆	CONJUNTOS DE TRABAJO	140
◆	CONTROLES DE ACCESO	473
◆	CRIPTOGRAFIA	479
■	D:	
◆	DESCRIPTOR DE ARCHIVOS	183
◆	DESEMPEÑO	301
◆	DIRECCIONAMIENTO	310
◆	DIRECTORIOS	156, 418
◆	DISCOS	210, 221, 223, 224
◆	DISEÑO	296
■	E:	
◆	EMBOTELLAMIENTOS	444
◆	ESCALABILIDAD	302, 433
◆	ESTACION DE TRABAJO	401
◆	ESTACIONES DE TRABAJO INACTIVAS	403
◆	ESTADO ESTABLE	452
◆	ESTADOS	039
◆	EVALUACION DEL RENDIMIENTO	438, 441
◆	EXCLUSION MUTUA	374

INDICE DE MATERIAS

■	F:	
◆	FALLOS	332, 434
◆	FLEXIBILIDAD	298
■	H:	
◆	HARDWARE	022, 271, 432
◆	HARDWARE DE E / S	198
◆	HILOS	395, 396
◆	HILOS Y RPC	400
■	I:	
◆	IMPLANTACION DE RPC	338
◆	INTERBLOQUEO	230
◆	INTERRUPCIONES	041
■	L:	
◆	LLAMADA A UN PROCEDIMIENTO REMOTO (RPC)	323
◆	LOCALIDAD	137
■	M:	
◆	MEDICIONES DEL RENDIMIENTO	439
◆	MEMORIA FIJA	032
◆	MODELACION	235
◆	MODELADO ANALITICO	446
◆	MODELO CLIENTE - SERVIDOR (C - S)	308, 320
◆	MODELOS	401, 406
◆	MULTICOMPUTADORAS	279, 280
◆	MULTIPROCESADOR	072
◆	MULTIPROCESADORES	078, 083, 273, 275
◆	MULTIPROCESAMIENTO	066, 076, 080

INDICE DE MATERIAS

◆	MULTIPROCESAMIENTO	
	SIMETRICO	081
◆	MULTIPROGRAMACION	093, 097, 101
■	N:	
◆	NUCLEO	044
◆	NUCLEOS DE SEGURIDAD	475
◆	NUMERO DE SERVIDORES	
	EN EL SISTEMA	449
■	O:	
◆	OBJETO	477
◆	OCULTAMIENTO	425
◆	OSTRICH	239
■	P:	
◆	PAGINA	143
◆	PAGINACION	110, 127, 142
◆	PAQUETE DE HILOS	397, 398
◆	PILA DE PROCESADORES	405
◆	PLANIFICACION	049, 051, 053, 056, 414
◆	PLANIFICACION DE	
	DISCOS	217, 218
◆	POISSON	448
◆	PRIMITIVAS	313, 315, 318
◆	PRIORIDADES	055
◆	PROBLEMAS EN RPC	347
◆	PROCESADOR	047
◆	PROCESADORES	394, 406, 407, 408, 409
◆	PROCESO DE POISSON	453
◆	PROCESOS	036, 045, 394
◆	PROCESOS DE MARKOV	460

INDICE DE MATERIAS

◆	PROCESOS DE NACIMIENTO Y MUERTE	461
◆	PROGRAMA	145
◆	PROTECCION	190
◆	PROTECCION POR CONTRASEÑA	472
◆	PROTOCOLOS	305
■	R:	
◆	RAM	225
◆	RECUPERACION	080
◆	RECURSOS	233
◆	REDES	433
◆	RELOJ DE INTERRUPCIONES	054
◆	RELOJES	225, 364, 368
◆	RENDIMIENTO	435, 436, 462
◆	REPLICA	429
◆	RESPALDO Y RECUPERACION	195
◆	RESULTADO DE LITTLE	452
◆	RETROALIMENTACION	445
◆	RPC	324, 328, 330
■	S:	
◆	SATURACION	444
◆	SEGMENTACION	120,127
◆	SEGURIDAD	184, 466, 467, 468
◆	SEGURIDAD EXTERNA	469
◆	SEGURIDAD OPERACIONAL	469

INDICE DE MATERIAS

◆ SEGURIDAD POR HARDWARE	475
◆ SINCRONIZACION	361, 362, 363, 370
◆ SISTEMA DE ARCHIVOS	149, 150, 161
◆ SISTEMA DISTRIBUIDO DE ARCHIVOS	422, 431, 432
◆ SISTEMA OPERATIVO	004, 482
◆ SISTEMAS DISTRIBUIDOS	266, 267, 269, 270, 303
◆ SISTEMAS OPERATIVOS	008, 011, 013
◆ SISTEMAS OPERATIVOS DE REDES	283
◆ SISTEMAS REALMENTE DISTRIBUIDOS	293
◆ SISTEMAS SUPERVIVIENTES	476
◆ SOFTWARE	028, 282
◆ SOFTWARE DE E / S	204

■ T:

◆ TENDENCIAS	021, 437
◆ TEORIA DE COLAS	447
◆ TERMINALES	226
◆ TIEMPO COMPARTIDO	294
◆ TIEMPOS DE SERVICIO	449
◆ TRAFICO	451
◆ TRANSACCION	381, 383
◆ TRANSACCIONES ATOMICAS	380
◆ TRANSPARENCIA	297
◆ TRATAMIENTO DEL BLOQUEO	264

INDICE DE MATERIAS

■	U:		
◆	USUARIOS MOVILES	434	
■	V:		
◆	VERIFICACION DE AMENAZAS	471	
◆	VIGILANCIA	471	

BIBLIOGRAFIA

- 01 - SISTEMAS OPERATIVOS MODERNOS:
 - ◆ TANENBAUM, ANDREW S.
- 02 - AN INTRODUCTION TO OPERATING SYSTEMS:
 - ◆ DEITEL, H. M.
- 03 - OPERATING SYSTEMS CONCEPTS:
 - ◆ PETERSON, J. L. & SILBERSCHATZ, A.
- 04 - THE LOGICAL DESIGN OF OPERATING SYSTEMS:
 - ◆ SHAW, A. C.
- 05 - OPERATING SYSTEMS:
 - ◆ MADNICK, S. E. & DONOVAN, J. J.
- 06 - FUNDAMENTALS OF OPERATING SYSTEMS:
 - ◆ LISTER, A. M.
- 07 - CONSTRUCCION DE SISTEMAS OPERATIVOS:
 - ◆ BORJA, JORGE L.
- 08 - ARQUITECTURAS RISC:
 - ◆ DEL PINO, G. & MARRONE, L. A.
- 09 - OPERATING SYSTEMS: DESIGN AND IMPLEMENTATION:
 - ◆ TANENBAUM, A. S.
- 10 - SISTEMAS OPERATIVOS DISTRIBUIDOS:
 - ◆ TANENBAUM, A. S.