

Índice general

Mensaje del editor	IX
Agradecimientos	XI
Sobre el autor	XIII
Prólogo	XXV
1. Introducción	1
1.1. Funciones y objetivos de los sistemas operativos	1
1.1.1. Funciones	3
1.2. Estructuras de los sistemas operativos	3
1.2.1. Gestión de procesos	3
1.2.2. Gestión de la memoria principal	4
1.2.3. Gestión de archivos	4
1.2.4. Gestión de dispositivos de entrada y salida	5
1.2.5. Servicios	5
1.2.6. Tipos de sistemas operativos	5
1.2.6.1. El núcleo del sistema operativo	8
1.3. Estructura y funcionamiento de la computadora	9
1.3.1. Componentes de una computadora	10
1.3.1.1. Dinámica de una operación de entrada y salida	10
1.3.1.2. El vector de interrupciones	11
1.3.1.3. Hardware de las interrupciones	11
1.3.1.4. Tratamiento de la interrupción	12

- 1.3.1.5. Tratamiento diferido 13
- 1.3.1.6. Clock tick 13
- 1.3.1.7. Llamadas al sistema 14
- 1.3.2. El proceso de arranque 16
- 1.4. Evolución de los sistemas operativos 17
- 1.4.1. La API de Windows 27
- 1.5. Trabajos prácticos 28
- 1.5.1. Práctica con Linux 28
- 1.5.1.1. procfs 29
- 1.5.1.2. Llamadas a sistema 30
- 1.5.1.3. Apagando Linux 31
- 1.5.2. Práctica con Windows 31
- 1.5.2.1. ProcFeatures 32
- 1.5.2.1.1. Ejemplo 1 32
- 1.5.2.1.2. Ejemplo 2 32
- 1.5.2.2. CoreInfo 33
- 1.5.2.2.1. Ejemplo 1 33
- 1.5.2.2.2. Ejemplo 2 33
- 1.5.2.3. Llamadas a sistema 33
- 1.5.2.4. Administrador de tareas 34
- 1.5.3. Una analogía 34
- 2. Procesos 35**
- 2.1. Estados de un proceso 37
- 2.1.1. Bloque de control de procesos 39
- 2.1.2. Creación y terminación de procesos 40
- 2.1.2.1. Jerarquía de procesos 41
- 2.1.3. Planificador y activador 41
- 2.1.4. Cambio de contexto 42
- 2.2. Procesos ligeros, hilos o threads 43
- 2.2.1. Motivaciones para el uso de hilos 45
- 2.2.2. Modelos de organización 48
- 2.2.3. Aspectos del diseño de hilos 49
- 2.2.3.1. Creación de los hilos 49
- 2.2.3.2. Terminación de los hilos 49
- 2.2.3.3. Sincronización de los hilos 50
- 2.2.3.4. Planificación de los hilos 50
- 2.2.3.5. Manejo de señales 51
- 2.2.4. Implementación 52
- 2.2.4.1. Ventajas y desventajas de los dos enfoques 53
- 2.2.5. Fibras 54
- 2.3. Comunicación 55
- 2.3.1. Procesos independientes 55
- 2.3.2. Procesos cooperantes 55
- 2.3.3. Procesos concurrentes 56

2.3.4.	Paradigmas de comunicación entre procesos	58
2.3.4.1.	Memoria compartida	58
2.3.4.2.	Pase de mensajes	58
2.3.4.3.	Puerto	60
2.3.4.4.	Señales	60
2.3.4.5.	Temporizadores	60
2.3.4.6.	Socket	61
2.3.4.7.	Tubería	61
2.3.4.8.	Cola de mensajes	61
2.3.4.9.	Archivos	61
2.3.4.10.	Archivos proyectados en memoria	61
2.3.5.	Excepciones	62
2.4.	Trabajos prácticos	62
2.4.1.	Práctica con Linux	62
2.4.1.1.	Estados de los procesos	62
2.4.1.2.	Obtener el PID	63
2.4.1.3.	Crear procesos con fork()	64
2.4.2.	Práctica con Windows	65
2.4.2.1.	tlist	65
2.4.2.1.1.	Ejemplo 1	66
2.4.2.1.2.	Ejemplo 2	66
2.4.2.2.	Información de los procesos con el Administrador de Tareas	67
3.	Memoria	69
3.1.	Funciones y operaciones del administrador de memoria	70
3.2.	Modelo de memoria de un proceso	71
3.2.1.	Fases en la generación de un ejecutable	72
3.2.2.	Formato del ejecutable	74
3.2.2.1.	El formato COFF	75
3.2.2.2.	El formato ELF	75
3.2.2.3.	El formato PE	76
3.2.3.	Mapa de memoria de un proceso	77
3.3.	Diferentes esquemas de administración	79
3.3.1.	Mono programación	79
3.3.1.1.	Partición absoluta única	79
3.3.1.2.	Partición reubicable única	80
3.3.1.3.	Superposiciones	81
3.3.2.	Multiprogramación	82
3.3.2.1.	Múltiples particiones fijas	82
3.3.2.2.	Múltiples particiones variables	84
3.3.2.2.1.	Algoritmos de selección de la partición:	85
3.3.2.2.2.	Cola de procesos por partición:	86
3.3.2.2.3.	Cola única:	86
3.3.2.3.	Sistema de compañeras	86
3.3.2.3.1.	Slab Allocator	86

3.3.2.4.	Reubicación	87
3.3.3.	Paginación simple	88
3.3.4.	Segmentación simple	90
3.3.5.	Segmentación con paginación	92
3.3.6.	Tablas de páginas y de segmentos	92
3.3.7.	Memoria asociativa	93
3.3.8.	Tabla de páginas invertida	94
3.3.9.	Ventaja adicional del paginado: las páginas compartidas	95
3.3.10.	Intercambio	96
3.4.	Memoria virtual	97
3.4.1.	Paginación por demanda	99
3.4.1.0.1.	Ejemplo	100
3.4.2.	Localidad de referencia	101
3.4.3.	Traba de páginas	102
3.4.4.	Tamaño de la página	103
3.4.5.	Algoritmos de reemplazo de páginas	103
3.4.6.	El rendimiento de los algoritmos	104
3.4.7.	Políticas de asignación	105
3.4.7.0.2.	Ejemplo	106
3.4.7.1.	Hiperpaginación	107
3.4.8.	Conjunto de trabajo	107
3.4.9.	Prepaginado	107
3.4.10.	Segmentación	108
3.5.	Trabajos prácticos	108
3.5.1.	Práctica con Linux	108
3.5.1.1.	Estadísticas de la memoria virtual en Linux	108
3.5.1.2.	El programa <code>mapa.c</code>	110
3.5.1.3.	La estructura de ELF	110
3.5.1.4.	El archivo <code>/proc/pid/maps</code>	110
3.5.2.	Práctica con Windows	112
3.5.2.1.	Memoria virtual en Windows	112
3.5.2.2.	La estructura de PE	113
3.5.2.3.	PhysMem	113
3.5.2.4.	CacheSet	114
4.	Planificación	115
4.1.	Conceptos	115
4.1.1.	Supuestos subyacentes	116
4.1.2.	Características	116
4.1.3.	Contexto	118
4.1.4.	Desalojo	118
4.1.5.	Colas	120
4.1.6.	Criterios	121
4.1.7.	Notación de Landau	122
4.2.	Tipos de planificación	122

4.3.	Algoritmos de planificación	124
4.3.1.	FCFS (First-Come, First-Serve)	124
4.3.1.1.	El efecto convoy	124
4.3.2.	SJF (Shortest-Job-First)	125
4.3.2.1.	SRPT (Shortest-Remaining-Processing-Time)	125
4.3.3.	Por prioridad	126
4.3.4.	RR (Round-Robin)	126
4.3.5.	Colas multiniveles	127
4.3.5.1.	MLQ	127
4.3.5.2.	MLFQ	127
4.4.	Planificación en POSIX	129
4.5.	Planificación en Linux	130
4.5.1.	Primeras versiones	130
4.5.1.1.	Version 0.96	130
4.5.1.2.	Versión 1.2	131
4.5.1.3.	Version 2.0.33	131
4.5.1.4.	Versión 2.2	132
4.5.1.5.	Versión 2.4	132
4.5.1.6.	Versión 2.6.8.1	133
4.5.2.	Versiones recientes	135
4.5.2.1.	Versión 2.6.21	135
4.5.2.2.	Versión 2.6.23	136
4.6.	Planificación en Windows	139
4.6.1.	Niveles de prioridad	140
4.6.2.	Base de datos del activador	142
4.7.	Práctica con Linux	143
4.7.1.	Cambio de contexto	144
4.7.2.	Una analogía	146
5.	Sincronización	147
5.1.	Antecedentes	147
5.2.	Alternativas de sincronización	148
5.2.1.	Sección crítica	149
5.2.2.	Exclusión mutua	150
5.2.3.	Soluciones hardware	150
5.2.3.1.	Inhabilitación de las interrupciones	150
5.2.3.2.	Instrucciones especiales de máquina	151
5.2.3.2.1.	TSL	151
5.2.3.2.2.	CAS	152
5.2.3.2.3.	Fetch-and-add	152
5.2.3.2.4.	Read-modify-write	152
5.2.3.2.5.	Load-link/store-conditional	152
5.2.3.2.6.	Soluciones libres de espera	152
5.2.4.	Soluciones software	153
5.2.4.1.	El problema de la sección crítica	153

5.2.4.1.1.	Algoritmo de Dekker	153
5.2.4.1.2.	Algoritmo de Peterson	153
5.2.4.1.3.	Requisitos de una solución	153
5.2.4.2.	Variables mutex	154
5.2.4.3.	Variables de condición	155
5.2.4.4.	Spinlock	156
5.3.	Problemas clásicos	156
5.3.1.	Los filósofos comensales	156
5.3.2.	El peluquero dormilón	157
5.3.3.	Problema del productor-consumidor	158
5.3.4.	Problema de los lectores-escritores	158
5.3.5.	Comunicación cliente-servidor	159
5.4.	Semáforos	159
5.5.	Monitores	161
5.5.1.	Sintaxis de un monitor	162
5.6.	Sincronización en pase de mensajes	162
5.6.1.	Características de la sincronización	163
5.6.2.	Memoria intermedia o <i>buffer</i>	165
5.6.2.1.	Buffer nulo	165
5.6.2.2.	Buffer de mensaje único	166
5.6.2.3.	Buffer de capacidad ilimitada	167
5.6.2.4.	Buffer de capacidad limitada	167
5.6.3.	Sincronización en tuberías	168
5.6.4.	Sincronización mediante señales	169
5.7.	Bloqueo mutuo	169
5.7.1.	Condiciones para que se cumpla el interbloqueo	170
5.7.2.	Grafos de asignación de recursos	170
5.7.3.	Métodos para el tratamiento del bloqueo mutuo	171
5.7.4.	Prevención	171
5.7.5.	Cómo evitar el bloqueo mutuo	172
5.7.5.1.	El algoritmo del banquero	173
5.7.6.	Detección del interbloqueo	173
5.7.7.	Recuperación del interbloqueo	174
5.7.8.	Algoritmo del avestruz	175
5.8.	Sincronización en POSIX	176
5.8.1.	Señales	176
5.8.2.	Mutex	177
5.8.2.1.	Mutex no bloqueante	178
5.8.3.	Semáforos para hilos	178
5.8.4.	Semáforos para procesos	179
5.8.5.	Variables de condición	179
5.8.6.	Memoria compartida	180
5.8.6.1.	Proyectar un archivo compartido	181
5.8.6.2.	Acceso compartido a un archivo	182
5.9.	Sincronización en Linux	183

5.9.1.	Operaciones atómicas	183
5.9.2.	Spinlock	184
5.9.2.1.	Full spinlock	184
5.9.2.2.	Lectores y escritores	184
5.9.3.	Mutex	185
5.10.	Sincronización en Windows	185
5.10.1.	Sincronización con IRQL alto	185
5.10.1.1.	Spinlocks	186
5.10.2.	Sincronización con IRQL bajo	187
5.10.3.	Llamada a Procedimiento Local	188
5.10.4.	Mailslots	190
5.11.	Trabajos prácticos	191
5.11.1.	Tuberías	191
5.11.2.	Otro ejemplo:	192
5.12.	Autoevaluación	193
6.	Almacenamiento	195
6.1.	Conceptos	196
6.1.1.	Archivos	197
6.1.1.1.	Tipos	197
6.1.1.2.	Atributos	198
6.1.1.3.	Operaciones	199
6.1.2.	Particiones	199
6.1.3.	Nombres	200
6.1.4.	Sistema de archivos	201
6.1.4.1.	Directorio	201
6.1.4.1.1.	Directorio de un nivel	202
6.1.4.1.2.	Directorio de dos niveles	202
6.1.4.1.3.	Estructura de árbol	203
6.1.4.1.4.	Estructura de grafo	203
6.1.4.1.5.	Entradas de directorio	204
6.1.4.2.	Designación de un archivo	206
6.1.4.2.1.	Directorio raíz por proceso	206
6.1.4.2.2.	Otros tipos de objetos	207
6.1.4.3.	Journaling	207
6.1.5.	Arquitectura del software	207
6.1.6.	Estructuras asociadas	209
6.1.7.	Organización de los archivos	211
6.1.7.1.	Gestión del espacio libre en disco	212
6.1.7.2.	Técnicas de asignación	213
6.2.	Implementaciones	216
6.2.1.	Sistemas de archivos en Windows	216
6.2.1.1.	FAT - File Allocation Table	216
6.2.1.2.	NTFS	220
6.2.2.	Sistema de archivos de UNIX	222

6.2.3.	Sistemas de archivos en Linux	225
6.2.3.1.	Second Extended File System	225
6.2.3.2.	Virtual File System	226
6.2.3.3.	Third Extended File System	227
6.2.3.4.	ReiserFS	228
6.2.3.5.	Fourth Extended File System	229
6.2.3.6.	Sistemas de archivos sin almacenamiento	230
6.2.3.6.1.	procfs	230
6.2.3.6.2.	debugfs	230
6.2.3.6.3.	sysfs	231
6.2.3.6.4.	ramfs	231
6.3.	Trabajos prácticos	232
6.3.1.	Particiones y sistemas de archivos	232
6.3.2.	Directorios, archivos y enlaces	234
6.3.3.	Permisos	236
7.	Sistemas de entrada y salida	237
7.1.	Dispositivos de entrada y salida	237
7.2.	Organización de las funciones de entrada y salida	238
7.3.	Aspectos del diseño	241
7.3.1.	Objetivos	241
7.3.2.	Mecanismo y política	241
7.3.3.	Estructura lógica de las funciones	241
7.4.	Subsistema de entrada y salida del núcleo	242
7.4.1.	Planificación de la entrada y salida	243
7.4.2.	Almacenamiento intermedio	243
7.4.2.1.	Memoria intermedia nula	244
7.4.2.2.	Memoria intermedia sencilla	244
7.4.2.3.	Memoria intermedia doble	245
7.4.2.4.	Memoria intermedia circular	245
7.4.3.	Caché	246
7.4.4.	Spool	247
7.4.5.	Manejo de errores	247
7.4.6.	Estructuras de datos	247
7.5.	Planificación del acceso a discos	248
7.5.1.	Estructura de un disco	248
7.5.2.	Planificación del acceso al disco	249
7.5.3.	Distintos tipos de algoritmos de planificación	249
7.5.3.1.	FCFS	250
7.5.3.2.	SSTF	250
7.5.3.3.	SCAN	250
7.5.3.4.	C-SCAN	251
7.5.3.5.	LOOK	251
7.5.3.6.	El ascensor Linus	251
7.5.3.7.	El planificador con fecha límite	252

7.5.3.8.	El planificador de entrada y salida anticipatorio	253
7.5.3.9.	CFQ	255
7.5.4.	Manejadores de dispositivos	255
7.6.	Ventanas al núcleo	257
7.6.1.	Desventajas	260
7.6.2.	devfs	261
7.6.3.	Conexión en caliente	262
7.6.4.	El nuevo modelo de dispositivos de Linux	262
7.6.4.1.	Colectores, dispositivos y clases	263
7.6.5.	sysfs	263
7.6.6.	udev	266
7.6.7.	HAL en Linux	267
7.6.7.1.	DeviceKit	268
7.6.8.	D-Bus	268
7.6.8.1.	Netlink sockets	268
7.6.9.	Cómo funciona todo junto	269
7.7.	Trabajos prácticos	271
7.7.1.	¿Cómo creamos archivos especiales?	273
7.8.	Autoevaluación	274
8.	Multiprocesamiento	275
8.1.	Múltiples procesadores	275
8.2.	Aplicaciones paralelas	277
8.3.	Ley de Amdahl	279
8.4.	Taxonomía de Flynn	280
8.5.	Arquitecturas de múltiples procesadores	283
8.5.1.	Procesamiento simétrico y asimétrico	285
8.6.	Multiprocesadores simétricos	286
8.6.1.	Ventajas	286
8.6.2.	Organizaciones	287
8.6.3.	Afinidad	289
8.6.4.	Granularidad	289
8.6.5.	Paralelismo	290
8.6.5.1.	Independiente	290
8.6.5.2.	De grano grueso y muy grueso	290
8.6.5.3.	De grano medio	291
8.6.5.4.	De grano fino	291
8.6.6.	Serialización de datos	291
8.7.	Diseño	292
8.7.1.	Procesos o hilos simultáneos y concurrentes	293
8.7.2.	Planificación	293
8.7.2.1.	Asignación de hilos a procesadores	294
8.7.2.2.	Uso de la multiprogramación en procesadores individuales	294
8.7.2.3.	Expedición de un proceso	295
8.7.2.4.	Planificación de procesos	295

8.7.2.5.	Planificación de multiprocesadores	295
8.7.3.	Sincronización	299
8.7.3.1.	Estructuras de datos locales a la CPU	299
8.7.3.2.	Spinlocks	299
8.7.3.3.	Semáforos - mutexes	300
8.7.3.4.	Traba de lectura rápida	300
8.7.3.5.	Leer-Copiar-Actualizar	301
8.7.3.6.	Barrera	301
8.7.4.	Gestión de la memoria	301
8.7.4.1.	Arquitectura de Acceso Uniforme a Memoria (UMA)	302
8.7.4.2.	Arquitectura de Acceso No Uniforme a Memoria (NUMA)	302
8.7.4.3.	Arquitectura de Memoria de Sólo Caché (COMA)	303
8.7.5.	Coherencia y consistencia	303
8.7.5.1.	Ordenamiento y consistencia de la memoria	305
8.7.5.2.	Modelos de consistencia	306
8.7.5.3.	Modelo de consistencia estricta	307
8.7.5.4.	Modelo de consistencia secuencial	308
8.7.5.5.	Modelo de consistencia causal	309
8.7.5.6.	Modelo de consistencia PRAM	309
8.7.5.7.	Modelo de consistencia de procesador	310
8.7.5.8.	Modelo de orden de almacenamiento total	310
8.7.5.9.	Modelo de consistencia débil	311
8.7.5.10.	Modelo de consistencia de liberación	312
8.7.5.11.	Modelo relajado de consistencia de liberación	313
8.7.5.12.	Ordenamiento explícito en memoria	313
8.7.5.13.	Las barreras de memoria del hardware	314
8.8.	Multiprocesamiento en Windows	314
8.8.1.	La base de datos del activador	315
8.8.2.	Sincronización en el núcleo	316
8.8.3.	Gestión de la memoria	316
8.9.	Multiprocesamiento en Linux	318
8.9.1.	Gestión de la memoria	319

Bibliografía

321