

ARQUITECTURA DE SISTEMAS OPERATIVOS

CÓDIGO DE MATERIA: 16

Titular: Leandro Ezequiel Mascarello
2022



FACULTAD DE TECNOLOGÍA INFORMÁTICA

Contenido

GUÍA DE REVISIÓN CONCEPTUAL	3
Unidad 0 – Generalidades Del Funcionamiento De La Computadora	3
Unidad 1 - Introducción, Definición Y Evolución De Los Sistemas Operativos	3
Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria	6
Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos	8
Unidad 4 - Dispositivos Periféricos Y Técnicas De Gestión De Almacenamiento De Información	11
Unidad 5 - Virtualización Y Computación En La Nube	14
Unidad 6 – Compiladores	15
GUÍA DE TRABAJOS PRÁCTICOS	17
Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria	17
GUIA PRÁCTICA DEL TP I	17
Unidad 5 - Virtualización Y Computación En La Nube	19
GUIA PRÁCTICA DEL TP II	19
GUIA PRÁCTICA DEL TP III	20
Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos	20
Unidad 6 – Compiladores	20
GUIA PRÁCTICA DEL TP IV	20
GUÍA DE ABORDAJE BIBLIOGRÁFICO	28
Unidad 1 - Introducción, Definición Y Evolución De Los Sistemas Operativos	28
Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria	28
Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos	28
Unidad 4 - Dispositivos Periféricos Y Técnicas De Gestión De Almacenamiento De Información	28
Unidad 5 - Virtualización Y Computación En La Nube	28
Unidad 6 – Compiladores	28

GUÍA DE REVISIÓN CONCEPTUAL

Unidad 0 – Generalidades Del Funcionamiento De La Computadora

1. Enumere y describa las partes mínimas que componen a una computadora.
2. Explique cómo está compuesta conceptualmente la memoria, y como se puede utilizar para representar datos de más de un byte.
3. Enumere y detalle cuales son los registros visibles para el usuario, y aquellos de control y estado.
4. Describa el proceso de Fetch (búsqueda) y Execution (ejecución) de Instrucciones
5. Explique qué es una Interrupción.
6. Describa los 4 tipos generales de interrupciones.
7. Enumere los mínimos pasos de software y hardware que deben darse para poder manejar interrupciones.
8. Explique el concepto de Jerarquía de Memoria.
9. ¿Qué características diferencian a los distintos niveles de memoria?
10. ¿En qué consiste el concepto de Caché? ¿Qué beneficios y problemas conlleva?
11. ¿Cómo funciona el acceso a E/S mediante programa?
12. ¿Cómo funciona el acceso a E/S mediante Interrupciones?
13. ¿Cómo funciona el acceso a E/S mediante DMA?
14. Dado un hipotético microprocesador de 32 bits, que cuenta con instrucciones de 32 bits compuestas por dos campos. El primer byte contiene el código de operación y el resto contiene el operando, o la dirección al operando
 - a. ¿Cuántas instrucciones distintas se pueden tener?
 - b. ¿Cuál es el valor máximo de un operando local?
 - c. ¿Cuál es la capacidad máxima de memoria direccionable?

Unidad 1 - Introducción, Definición Y Evolución De Los Sistemas Operativos

15. ¿Qué es un sistema operativo?
16. ¿Cuáles son los objetivos de un S.O.?

17. ¿A quiénes y de qué forma da servicio un S.O.?
18. ¿Qué servicios típicamente se encuentran en un S.O.?
19. Enumere y describa qué situaciones pueden llevar a la necesidad de evolucionar un S.O.
20. Explique el modelo de procesamiento Serial. ¿Qué problemas tiene?
21. Explique el modelo de ejecución por lotes ¿Qué ventajas tiene?
22. ¿qué características de hardware son útiles para implementar un monitor para un sistema de ejecución por lotes?
23. Defina Multitasking.
24. ¿Qué es Time Sharing (tiempo compartido)?
25. ¿Cuándo es beneficioso utilizar planificación de multiprogramación por lotes y cuando tiempo compartido? ¿Por qué?
26. Enumere los 5 principales avances realizados por los sistemas operativos.
27. Enumere y describa las cuatro causas principales de errores de sincronización entre procesos.
28. ¿Cuáles son las responsabilidades del sistema operativo en cuanto a manejo de memoria?
29. ¿Cuáles son las responsabilidades del sistema operativo en cuanto a la seguridad de la información?
30. ¿Cuáles son las responsabilidades del sistema operativo en cuanto al manejo de recursos y agendamiento?
31. Nombre, detalle a que objetos refiere, y ejemplifique las funcionalidades propuestas por los niveles jerárquicos de sistemas operativos
32. ¿Qué es un proceso?
33. ¿Qué características suele tener un proceso?
34. Defina Bloque de Proceso.
35. ¿En qué condiciones se crean procesos?
36. Enumere y describa los pasos necesarios para crear un proceso.
37. ¿En qué condiciones se finalizan procesos?

38. ¿Qué estados y transiciones tiene un motor de procesos de 2 estados? ¿Qué problemas tiene?
39. ¿Qué estados y transiciones tiene un motor de procesos de 5 estados? ¿Qué problemas tiene?
40. Explique el funcionamiento básico que debe tener el sistema operativo para poder manejar procesos de 5 estados.
41. ¿Cuál es la diferencia entre el estado “Bloqueado” y el estado “Suspendido”?
42. ¿Qué componentes tiene la imagen de proceso?
43. ¿Qué modos de ejecución existen? ¿Qué los diferencia?
44. ¿Qué mecanismos existen para interrumpir la ejecución de un proceso? Ejemplifique
45. Enumere y describa los pasos necesarios para hacer un cambio del proceso en ejecución (Process Switch)
46. ¿Es el sistema operativo un conjunto de procesos? ¿En qué escenario?
47. Diagrame los 9 estados y las transiciones del motor de procesos de Unix
48. ¿Cómo es la imagen de proceso de Unix?
49. ¿Cómo es la creación de procesos en Unix?
50. Explique la diferencia entre Paralelismo y Concurrencia.
51. Defina Multithreading
52. Explique que tipos de sistemas pueden implementarse combinando la cantidad de procesadores, y la cantidad de threads (hilos) por proceso.
53. ¿Cuáles son los beneficios de tener threads (hilos)?
54. ¿Cuáles son los estados básicos de un thread?
55. ¿Qué tipos de thread se pueden encontrar? ¿Qué los diferencia?
56. ¿Qué beneficios tienen los threads (hilos) a nivel de usuario?
57. ¿Qué beneficios tienen los threads (hilos) a nivel de kernel (núcleo)?
58. ¿Qué es el multiprocesamiento simétrico (SMP)? ¿Qué posibles ventajas conlleva?
59. Enumere y describa las categorías de sistemas propuestas por Flynn
60. ¿Qué problemas deben tenerse en cuenta a la hora de diseñar un sistema operativo que funcione sobre un multiprocesador?

61. ¿Qué es un Microkernel?
62. ¿Qué llevó a la creación de los sistemas de tipo Microkernel?
63. ¿Cuáles son los beneficios de los sistemas de tipo Microkernel?
64. ¿Cuáles son las desventajas de los sistemas de tipo Microkernel?
65. ¿Qué características tienen los Procesos e threads (hilos) de Windows?
66. ¿Qué estados y transiciones tienen los threads (hilos) de Windows?
67. ¿Qué estados y transiciones tienen los threads (hilos) de Linux?
68. Enumere similitudes y diferencias entre Windows y Linux.

Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria

69. ¿Qué diferentes escalas de agendamiento de procesos se tienen?
70. ¿Qué define el agendamiento de largo plazo?
71. ¿Qué define el agendamiento de mediano plazo?
72. ¿Qué define el agendamiento de corto plazo?
73. ¿Qué situaciones pueden causar un nuevo agendamiento de corto plazo?
74. Enumere y explique los criterios de agendamiento orientados al usuario
75. Enumere y explique los criterios de agendamiento orientados al sistema
76. Explique el algoritmo de First Come First Served (FCFS). ¿qué características tiene?
77. Explique el algoritmo de Round Robin (RR). ¿qué características tiene?
78. Explique el algoritmo de Shortest Process Next (SPN). ¿qué características tiene?
79. Explique el algoritmo de Shortest Remaining Time (SRT). ¿qué características tiene?
80. Explique el algoritmo de Highest Response Ratio Next(HRRN). ¿qué características tiene?
81. Explique el algoritmo de Feedback. ¿qué características tiene?
82. ¿Qué modos de decisión existe? ¿Cómo difieren?
83. Explique el concepto de inanición

84. ¿cómo asigna procesos tradicionalmente Unix?
85. Describa los diferentes tipos de multiprocesador que se pueden encontrar
86. ¿Qué es la granularidad?
87. Describa los distintos niveles de granularidad
88. ¿Qué implica una granularidad fina?
89. ¿Qué implica una granularidad gruesa?
90. ¿En que escenario pueden coexistir distintas granularidades? Ejemplifique.
91. ¿Qué problemas deben tenerse en cuenta para asignar procesos en un multiprocesador?
92. Explique el problema de asignación de un proceso a un procesador. ¿Qué variantes existen?
93. ¿Qué problema surge con utilizar un programa concurrente en un solo procesador?
94. Enumere y defina los algoritmos de asignación de Threads (hilos) en multiprocesadores
95. Explique el método de Load Sharing. ¿Qué beneficios tiene?
96. Explique el método de Gang Scheduling. ¿Qué beneficios tiene?
97. Explique el método de Dedicated Processor Assignment. ¿Qué beneficios tiene?
98. Explique el método de Dynamic Scheduling. ¿Qué beneficios tiene?
99. ¿Qué es un sistema operativo de tiempo real?
100. ¿Qué caracteriza a los sistemas de tiempo real?
101. ¿cómo funciona el agendamiento en un sistema operativo de tiempo real?
102. Explique el algoritmo de aproximación estática orientado a tabla para asignación en tiempo real
103. Explique el algoritmo de aproximación estática orientado a prioridad preventiva para asignación en tiempo real
104. Explique el algoritmo de aproximación dinámica orientado a planes para asignación en tiempo real
105. Explique el algoritmo de aproximación dinámica del mejor esfuerzo para asignación en tiempo real
106. Explique el algoritmo de agendamiento de Deadline (fecha limite)

- 107. Explique el algoritmo de agendamiento de tasa monotónica
- 108. Explique el problema de inversión de prioridad. ¿En qué caso puede ocurrir?
- 109. ¿Qué tipos de agendamiento en tiempo real maneja Linux?
- 110. Compare a Windows con Linux en cuanto a agendamiento de procesos

Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos

- 111. ¿Qué es una operación atómica?
- 112. ¿qué es una sección crítica?
- 113. ¿Qué dificultades existen a la hora de trabajar con sistemas concurrentes?
- 114. ¿De qué sirve el interleaving (intercalado)?
- 115. ¿Qué es una Race Condition?
- 116. ¿Qué forma de interacción entre los procesos existe?
- 117. ¿Qué beneficios puede traer tener la capacidad de desactivar las interrupciones?
- 118. ¿Qué es un Semáforo?
- 119. Describa qué operaciones se pueden realizar en un Semáforo
- 120. ¿Qué es un Mutex? ¿Qué lo diferencia de un Semáforo?
- 121. ¿Qué es un Spinlock?
- 122. ¿Qué es un Monitor? Ejemplifique
- 123. ¿Qué beneficios tiene el modelo de pasaje de mensajes?
- 124. ¿Qué tipos de pasaje de mensajes existen?
- 125. ¿Qué tipos de recursos se tiene? ¿Qué implicancias conllevan?
- 126. Explique el fenómeno de Deadblock (interbloqueo)
- 127. ¿Qué condiciones deben darse para que surja un Deadlock (interbloqueo)?
- 128. ¿Qué condiciones se deben dar para que exista un Deadlock (interbloqueo)?
- 129. ¿Qué acciones pueden tomarse para resolver un Deadlock (interbloqueo)?
- 130. ¿Cómo se puede detectar un Deadlock (interbloqueo)?
- 131. ¿Qué acciones se pueden tomar para prevenir un Deadlock (interbloqueo)?

132. Detalle el escenario de la Cena de los Filósofos. ¿Cómo se podría solucionar?
133. Explique el concepto de las Barreras como instrucción del procesador ¿En qué caso son necesarias?
134. Explique el concepto de Pipes (cañerías) como mecanismo de concurrencia de Unix
135. Explique el concepto de Mensajes como mecanismo de concurrencia de Unix
136. Explique el concepto de Memoria compartida como mecanismo de concurrencia de Unix
137. Explique el concepto de Semáforos como mecanismo de concurrencia de Unix
138. Explique el concepto de Señales como mecanismo de concurrencia de Unix
139. Compare Windows con Linux en cuanto a su manejo de Concurrencia
140. ¿Cuáles son las funciones que debe proveer el manejo de memoria?
141. Defina Relocation (reubicación) en el contexto de manejo de memoria
142. Defina Protección en el contexto de manejo de memoria
143. Defina compartir en el contexto de manejo de memoria. ¿Cómo interactúa con la necesidad de seguridad?
144. Explique que son la organización Física y Lógica de la memoria
145. Explique la técnica de Particionamiento fijo de la memoria. ¿Qué ventajas y desventajas tiene?
146. ¿Qué impacto tiene el tamaño de las particiones en el particionamiento fijo de memoria?
147. ¿Cómo es el algoritmo de asignación para el particionamiento fijo de memoria?
148. Explique la técnica de Particionamiento dinámico de la memoria. ¿Qué ventajas y desventajas tiene?
149. ¿Cómo es el algoritmo de asignación para el particionamiento dinámico de memoria?
150. ¿Qué complejidad extra tiene la asignación de partes dinámicas por sobre las fijas? ¿Cómo se resuelve?
151. Explique la técnica de paginación simple de la memoria. ¿Qué ventajas y desventajas tiene?
152. ¿Qué mejoras provee la paginación por sobre la partición? ¿Por qué?

153. Explique la técnica de Segmentación simple de la memoria. ¿Qué ventajas y desventajas tiene?
154. Explique la técnica de Memoria virtual paginada. ¿Qué ventajas y desventajas tiene?
155. Explique la técnica de Memoria virtual segmentada. ¿Qué ventajas y desventajas tiene?
156. ¿Qué beneficios tiene combinar la paginación y la segmentación?
157. ¿Qué es la memoria virtual?
158. ¿Qué costos tiene el utilizar la memoria virtual?
159. ¿Qué beneficios tiene utilizar la memoria virtual?
160. ¿Cómo es la mínima estructura de control necesaria para manejar memoria paginada?
161. ¿Cómo es la mínima estructura de control necesaria para manejar memoria Segmentada?
162. ¿Cómo es la mínima estructura de control necesaria para manejar memoria paginada y segmentada?
163. ¿Cómo es el proceso de traducción de direcciones de un sistema paginado?
164. ¿Qué consideraciones hay que tener para la protección y el compartir segmentos de memoria?
165. ¿Qué es una falla de paginado? ¿Qué costo tiene?
166. ¿Cómo funciona el algoritmo óptimo de reemplazo de páginas?
167. ¿Cómo funciona el algoritmo de uso menos reciente (LRU) de reemplazo de páginas?
168. ¿Cómo funciona el algoritmo primero en llegar, primero en salir (FIFO) de reemplazo de páginas?
169. ¿Cómo funciona el algoritmo de reloj (Clock) de reemplazo de páginas?
170. ¿Qué variantes de definición de tamaño residente existen? ¿Qué ventajas poseen?
171. ¿Qué caracteriza al manejo de memoria de Unix?
172. ¿Cómo son las estructuras de datos del manejo de páginas virtuales de Unix?
173. ¿Cómo es el algoritmo de reemplazo de páginas de Unix?

174. ¿Qué diferencia al manejo de memoria de Linux con el de Unix?
175. ¿Qué diferencia al manejo de memoria de Windows con el de Linux?

Unidad 4 - Dispositivos Periféricos Y Técnicas De Gestión De Almacenamiento De Información

176. Describa en qué categorías se pueden agrupar los dispositivos de E/S
177. Enumere y defina las características que diferencian a las distintas categorías de dispositivos E/S
178. Describa a grandes rasgos como evolucionaron los dispositivos de E/S
179. Describa el proceso de Direct Memory Access (DMA)
180. ¿Cuál es el mínimo hardware necesario para realizar Direct Memory Access (DMA)?
181. ¿Qué beneficios trae hacer Direct Memory Address (DMA)?
182. ¿Cuáles son los objetivos de diseño a tener en cuenta a la hora de construir sistemas de E/S?
183. Diferencie el concepto E/S Lógica de E/S de dispositivo
184. ¿Qué es un File System (sistema de archivos)?
185. ¿Qué es el manejo de Directorios?
186. Explique el concepto de Buffer. ¿Qué beneficios y costos conlleva?
187. ¿Qué es un Buffer doble? ¿Qué beneficios trae por sobre el simple y a qué costo?
188. ¿Qué es un Buffer Circular? ¿Qué beneficios trae por sobre los otros tipos?
189. Explique el concepto de Seek Time (Tiempo de búsqueda). ¿Qué características físicas lo impactan?
190. Defina Tiempo de Transferencia.
191. ¿Cómo se almacena información en un disco duro tradicional?
192. ¿Cómo se accede a la información en un disco duro tradicional?
193. ¿Qué diferencias estructurales tiene un disco en estado sólido? ¿Qué beneficios trae?
194. Explique el algoritmo First in First Out (FIFO) de Planificación de acceso a disco.

195. Explique el algoritmo Last in First Out (LIFO) de Planificación de acceso a disco. ¿Presenta una mejora por sobre FIFO?
196. Explique el algoritmo Shortest Service Time First (SSTF) de Planificación de acceso a disco. ¿En qué escenario es mejor utilizarlo?
197. Explique el algoritmo SCAN de Planificación de acceso a disco. ¿Qué ventaja presenta contra los anteriores?
198. Explique el algoritmo C-SCAN de Planificación de acceso a disco. ¿Qué ventaja presenta por sobre SCAN?
199. ¿Qué problema dio lugar al surgimiento de F-SCAN y N-Step-Scan? ¿Cómo lo solucionan?
200. ¿Qué significan las siglas RAID?
201. Explique el concepto de RAID
202. ¿Qué posibles beneficios tiene la técnica de RAID?
203. Describa a la configuración RAID 0. ¿Qué beneficios trae?
204. Diagrame el estado de los discos para RAID 0
205. Describa a la configuración RAID 1. ¿Qué beneficios trae?
206. Diagrame el estado de los discos para RAID 1
207. Describa a la configuración RAID 2. ¿Qué beneficios trae?
208. Diagrame el estado de los discos para RAID 2
209. Describa a la configuración RAID 3. ¿Qué beneficios trae?
210. Diagrame el estado de los discos para RAID 3
211. Describa a la configuración RAID 4. ¿Qué beneficios trae?
212. Diagrame el estado de los discos para RAID 4
213. Describa a la configuración RAID 5. ¿Qué beneficios trae?
214. Diagrame el estado de los discos para RAID 5
215. Describa a la configuración RAID 6. ¿Qué beneficios trae?
216. Diagrame el estado de los discos para RAID 6
217. Explique el concepto de RAID por Software. ¿Qué beneficios trae?
218. ¿Qué estrategias de reemplazo se utilizan en el caché del disco?

219. Describa cómo Linux planifica sus accesos a disco
220. Compare el acceso a dispositivos de E/S de Linux y Windows
221. ¿Qué es un File System (sistema de archivos)?
222. ¿Qué propiedades debe tener un File System (sistema de archivos)?
223. ¿Qué operaciones se pueden realizar sobre un File System (sistema de archivos)?
224. Defina el concepto de Field (Campo)
225. Defina el concepto de Record (Registro)
226. Defina el concepto de File (Archivo)
227. Defina el concepto de Database (Base de datos)
228. ¿Qué es un File Management System (Sistema de Manejo de Archivos)?
229. ¿Qué objetivos tiene un File Management System (Sistema de Manejo de Archivos)?
230. ¿Cómo interactúan entre sí el File Management System (Sistema de Manejo de Archivos) con el Sistema operativo?
231. ¿Qué criterios impactan a la hora de decidir qué estructura lógica de almacenamiento se utilizará?
232. Explique como funciona un archivo de tipo Pile (montón).
233. Explique como funciona un archivo secuencial. ¿Qué beneficios trae sobre el anterior?
234. Explique como funciona un archivo secuencial indexado. ¿Qué beneficios trae sobre el anterior?
235. Explique como funciona un archivo indexado. ¿Qué beneficios trae sobre el anterior?
236. Explique como funciona un archivo del tipo Hashed (directo). ¿Qué beneficios trae?
237. Compare las velocidades de acceso y actualización de los distintos tipos de archivo
238. ¿Qué es un Directorio?
239. ¿Qué información tiene un Directorio?
240. ¿Qué problemas surgen a la hora de reservar espacio para archivos?
241. Explique el problema de la Fragmentación. ¿En qué escenario puede darse?

- 242. ¿Cómo se resuelve el problema de la Fragmentación?
- 243. Explique el algoritmo de reserva de espacio de reserva contigua. ¿cómo es la tabla de archivos?
- 244. Explique el algoritmo de reserva de espacio de reserva encadenada. ¿cómo es la tabla de archivos? ¿Qué beneficios y costos conlleva?
- 245. Explique el algoritmo de reserva de espacio de reserva indexada. ¿cómo es la tabla de archivos? ¿Qué beneficios y costos conlleva?
- 246. ¿Qué es un Volumen?
- 247. ¿Qué tipos de archivos maneja Unix?
- 248. ¿Cómo reserva el espacio Unix?
- 249. ¿Qué es el Virtual File System de Linux? ¿Qué beneficios tiene?
- 250. ¿Qué caracteriza al formato FAT?
- 251. ¿Qué caracteriza al formato NTFS?
- 252. Compare como manejan los archivos Linux y Windows

Unidad 5 - Virtualización Y Computación En La Nube.

- 253. ¿Qué beneficios trae el procesamiento distribuido?
- 254. ¿Qué es un Cluster?
- 255. Defina Middleware.
- 256. ¿Qué es la emulación?
- 257. ¿Qué es una máquina virtual?
- 258. Explique qué caracteriza a una máquina Virtual de sistema
- 259. Explique qué caracteriza a una máquina virtual de proceso
- 260. ¿Cómo funciona la virtualización total por Binary Rewriting?
- 261. ¿Cómo funciona la paravirtualización? ¿Qué beneficios trae?
- 262. ¿Cómo funciona la virtualización asistida por hardware? ¿Qué beneficios trae?
- 263. ¿Qué es un Hipervisor?
- 264. ¿Qué diferencia a los hipervisores de tipo 1 y 2?

- 265. ¿Qué es la “Cloud” a la que se referencia con Cloud Computing?
- 266. ¿Qué beneficios trae el uso de Cloud?
- 267. Defina Escalabilidad (vertical)
- 268. Defina Elasticidad (horizontal)
- 269. Explique el concepto de SaaS (Software as a Service)
- 270. Explique el concepto de PaaS (Platform as a Service)
- 271. Explique el concepto de IaaS (Infrastructure as a Service)
- 272. ¿Qué es el balanceo de cargas? Ejemplifique
- 273. Explique cómo funciona un CDN (Content Distribution Network)

Unidad 6 – Compiladores

- 274. ¿Qué es código ejecutable?
- 275. ¿Qué es un Lenguaje de Programación?
- 276. ¿Qué es un Abstract syntax tree (AST)?
- 277. ¿Qué es un Compilador?
- 278. ¿Qué caracteriza a un multi-pass Compiler?
- 279. ¿Qué fases tiene el proceso de compilación?
- 280. ¿Qué actividades se realizan en la fase de compilación Front-End?
- 281. ¿Qué actividades se realizan en la fase de compilación Middle-End?
- 282. ¿Qué actividades se realizan en la fase de compilación Back-End?
- 283. ¿Qué es un Assembler?
- 284. ¿Qué beneficios tiene el uso de compiladores?
- 285. ¿Qué particularidad tiene un Cross Compiler?
- 286. ¿Qué es un Interpreter?
- 287. ¿Qué caracteriza a un interpreter de bytecodes?
- 288. ¿Qué caracteriza a un threaded interpreter?
- 289. ¿Qué caracteriza a un interpreter de AST?

- 290. ¿Qué es el microcode?
- 291. ¿Qué es un Linker?
- 292. ¿Cómo funciona el linking dinámico? ¿Qué beneficios trae?
- 293. ¿Cómo funciona el linking estático? ¿Qué beneficios trae?
- 294. Explique el funcionamiento básico de un Compilador
- 295. Explique el funcionamiento básico de un interpreter
- 296. Explique qué es un Just-in-time Compiler. ¿Qué beneficios trae?

GUÍA DE TRABAJOS PRÁCTICOS

Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria

GUIA PRÁCTICA DEL TP I

El objetivo de este trabajo es brindar a los alumnos con la posibilidad de aplicar los conceptos teóricos abordados por la asignatura.

Los alumnos, en equipos de no más de dos personas, deberán analizar la evolución de la ejecución de un conjunto de procesos simplificados en una serie de procesadores ideales, y detallar los resultados de estos análisis en una carpeta entregable.

Dados las siguientes cargas de trabajo compuestas, cada una, por un conjunto de procesos definidos por:

1. Nombre (P1, P2, P3, etc.)
2. Esfuerzo (el número de ciclos de ejecución que completarán este trabajo)
3. Prioridad (Número positivo que indicará la prioridad de este proceso. A menor valor, mayor prioridad)
4. Tiempo de arribo (Número positivo que indicará en que ciclo de ejecución el proceso será disparado)

Para cada carga de trabajo los alumnos deberán:

- A. Detallar en forma narrativa o gráfica la evolución de la ejecución de cada escenario
- B. Identificar las siguientes métricas en cada escenario
 - a. Tiempo total necesitado para completar todos los procesos
 - b. Cantidad de tiempo que el procesador perdió realizando cambios de contexto
 - c. Tiempo máximo, mínimo y promedio de espera de los procesos agrupados por prioridad
 - d. Tiempo máximo, mínimo y promedio de espera de los procesos en general
- C. Realizar un cuadro comparativo de los distintos escenarios con respecto a las métricas realizadas
- D. Escribir una conclusión donde analicen el desempeño de los distintos algoritmos ante las distintas cargas de trabajo

Escenarios

Nombre	Algoritmo de Planificación	Costo de cambio de contexto
FCFS	First Come- First Served.	No hay cambio de contexto
SJF – N	Shortest-Job-First, sin interrupciones	No hay cambio de contexto
SJF – P0	Shortest-Job-First, con interrupciones	0 ciclos de procesador
SJF – P1	Shortest-Job-First, con interrupciones	1 ciclo de procesador
SJF – P3	Shortest-Job-First, con interrupciones	3 ciclos de procesador
RR – 1 0	Round Robin con ventanas de ejecución de 1 ciclo de procesador	0 ciclos de procesador

RR – 1 1	Round Robin con ventanas de ejecución de 1 ciclo de procesador	1 ciclo de procesador
RR – 2 1	Round Robin con ventanas de ejecución de 2 ciclo de procesador	1 ciclo de procesador
Priority – N	Colas múltiples con prioridades, sin interrupciones	No hay cambio de contexto
Priority – P 1	Colas múltiples con prioridades, con interrupciones	1 ciclo de procesador

Cargas de trabajo

Carga de trabajo 1

Nombre	Esfuerzo	Prioridad	Tiempo de Arribo
P1	5	1	0
P2	5	1	4

Carga de trabajo 2

Nombre	Esfuerzo	Prioridad	Tiempo de Arribo
P1	5	5	0
P2	5	1	4

Carga de trabajo 3

Nombre	Esfuerzo	Prioridad	Tiempo de Arribo
P1	50	1	0
P2	20	2	10
P3	40	3	6

Carga de trabajo 4

Nombre	Esfuerzo	Prioridad	Tiempo de Arribo
P1	50	9	0
P2	20	4	30
P3	1	3	40
P4	1	2	40

Carga de trabajo 5

Nombre	Esfuerzo	Prioridad	Tiempo de Arribo
P1	2	9	0
P2	2	8	1
P3	2	7	2
P4	2	6	3

P5	2	5	4
P6	2	4	5
P7	2	3	6

El docente encargado del curso evaluará la posibilidad de la exposición por parte de los alumnos, en clase y/o en el laboratorio de computación, de acuerdo a la disponibilidad de recursos. De estas exposiciones el curso podrá apreciar en la práctica como distintos algoritmos de asignación pueden resultar preferibles ante distintas cargas de trabajo, y generará un consenso sobre el trabajo realizado.

Unidad 5 - Virtualización Y Computación En La Nube.

GUIA PRÁCTICA DEL TP II

El objeto de este trabajo es familiarizar a los estudiantes con el uso de un software de virtualización, máquinas virtuales, y los comandos básicos de un sistema operativo en particular. A través de exposiciones dicho conocimiento se compartirá entre los estudiantes para lograr una visión generalizada y un consenso en el curso.

Los alumnos, en equipos de no más de cuatro personas, deberán realizar una carpeta entregable, y una presentación donde muestren el resultado del desarrollo de los siguientes puntos en función de un sistema operativo en particular para cada alumno designado por el docente:

1. Investigar un software operador de máquina virtual, instalarlo y cargar un sistema operativo en él. El mercado ofrece distinto software para ese cometido como ser VIRTUALBOX, VMWARE u otros. Queda a discreción de los alumnos la selección del software de virtualización en función de sus preferencias, experiencias previas, etc.
2. Crear una máquina virtual donde los alumnos deberán instalar el sistema operativo en la versión específica que fue designada por el docente.
3. Los alumnos deberán poder mostrar a dicha máquina virtual funcionando, demostrando que cuentan con
 - a. Acceso al sistema de archivos y persistencia
 - b. Acceso a internet
 - c. Capacidad de instalar y ejecutar aplicaciones
4. Los alumnos deberán analizar y demostrar para el sistema que les fue asignado el uso, configuración y comandos si correspondiere de:
 - a. Arquitectura del sistema operativo y sus características generales
 - b. Administración de usuarios
 - i. Creación, modificación y eliminación de usuarios
 - ii. Perfiles de usuarios, grupos y permisos
 - c. Administración de recursos
 - i. Directorios y archivos
 - ii. Copias de resguardo

- d. Seguridad
 - i. Aspectos relevantes
- e. Correo electrónico
 - i. Protocolos de comunicaciones
 - ii. Protocolo de transferencia de archivos

El docente encargado del curso evaluará la posibilidad de la exposición por parte de los alumnos, en clase y/o en el laboratorio de computación, de acuerdo con la disponibilidad de recursos.

GUIA PRÁCTICA DEL TP III

El objeto de este trabajo es familiarizar a los estudiantes con el uso de Cloud Computing, basándose en su práctica previa con máquinas virtuales. Para ello los alumnos desarrollarán un análisis sobre las ofertas y viabilidades de Cloud Computing, y montarán su propia máquina virtual en un servidor Cloud. Este trabajo generará como entregable una carpeta y una presentación que los alumnos realizarán para el resto del curso.

Los alumnos deberán realizar una investigación sobre Cloud Computing en el mercado, e implementarán su propia máquina virtual en el contexto de Cloud.

Para ello se seguirán los siguientes pasos:

1. Buscar proveedores de Cloud Computing, como oferta de servidores para trabajos de empresas como ser: procesadores, disponibilidad de espacio de almacenamiento masivo, servicios de back up, etc.
2. Realizar un análisis comparativo de las variantes en servicio, prestaciones, y costos para los proveedores seleccionados.
3. Obtener infraestructura de un proveedor de Cloud, o montar su propia infraestructura utilizando software libre (como por ejemplo OpenStack, CloudStack, OpenNebula).
4. Instalar en dicha infraestructura la máquina virtual desarrollada anteriormente y demostrar su funcionamiento.
5. Realizar un análisis de beneficios y perjuicios de montar su sistema sobre infraestructura en la nube.

Los alumnos deberán preparar un informe entregable sobre el proceso, y demostrar el funcionamiento de sus máquinas virtuales en el aula.

El docente encargado del curso evaluará la posibilidad de la exposición por parte de los alumnos, en clase y/o en el laboratorio de computación, de acuerdo con la disponibilidad de recursos, y definirá si lo considera propicio que servicio de Cloud utilizará cada alumno.

Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos

Unidad 6 – Compiladores

GUIA PRÁCTICA DEL TP IV

La idea del siguiente trabajo es que tenga un orden didáctico y sencillo para que el alumno pueda realizar sus propias modificaciones ó realizar un intérprete completamente radical.

El documento muestra cómo implementar/agregar un nuevo shell ó interfaz de comandos en un sistema operativo del tipo Unix-Linux, pudiendo existir variaciones en su implementación de acuerdo a la distribución del sistema operativo elegido.

MAGNITUD (entendimiento de su naturaleza por www.cs.cinvestav.mx)

Un shell es un intérprete de comandos, una interfaz entre el usuario y el SO. Es algo que nos provee de un entorno que nos permite introducir comandos en el sistema y que éste los reconozca. Linux tiene diferentes tipos de Shells: Bourne (sh), Bourne Again SHell (BASH), Korn (ksh), C shell (csh)...

- BourneSh: es la que se utilizaba originalmente en los sistemas UNIX y está disponible en todos los sistemas UNIX existentes. Es muy adecuada para programación de shell, pero no se comporta demasiado bien de cara al usuario.
- CSh se comporta mejor ante el usuario y tiene algunas características que no están presentes en Bourne, como el autocompletado de línea, pero no es tan buena para programación. Su sintaxis es parecida a la del lenguaje C.
- Korn unió lo mejor de las dos anteriores, manteniendo la compatibilidad con la primera.

Actualmente no se usan mucho las shells antes mencionados, sino otras que son evolución de aquellas y que tienen características de las tres. Estas son bash (Bourne again), pdksh (Public Domain Korn Shell) y tesh (una extensión de csh). En la actualidad por defecto la mayoría de las distribuciones usan el BASH. No obstante algunos en algunos entornos se suele desarrollar programas específicos para tener controlado lo que el usuario puede realizar, estos programas pueden ser scripting o compilados.

Ejemplo de Shell (vía script)

```
#!/bin/sh
while true; do
    clear
    echo "ingrese palabra clave"
    read aux
    if [ ${aux} = "listar" ] ; then
        find /
    fi
done
exit 0
```

La primer línea es de suma importancia porque indica que interprete se debe usar para leer el script, en este caso es también un shell script por simplicidad, pero tranquilamente se

podría haber usada un lenguaje de scripting de mayor poder como por ejemplo perl ó phyton.

El script hace un bucle infinito que la única forma de cortarlo es vía Control+C, aunque con la ultima línea hace que finalice y se cierre la terminal por la cual se accedió al sistema (consola, telnet ó ssh).

En ese bucle se espera que se ingrese únicamente la palabra “listar” para así obtener un listado de todos los archivos del sistema con el comando find.

Permisos

El ejemplo anterior le hemos puesto el nombre de “test.sh”, este archivo se forma a traves comandos preexistentes para realizar un script, a su vez lo hemos guardado en el directorio /bin con permisos de ejecución como se denota debajo.

```
[root@ns3 ~]# chmod 755 /bin/test.sh
```

```
[root@ns3 ~]# ll /bin/test.sh
-rwxr-xr-x 1 root root 127 Apr 16 20:54 /bin/test.sh
```

Una vez realizado esto hay que agregarlo en el archive /etc/shells para poder ser invocado y así asignarlo a un usuario

```
[root@ns3 ~]# cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/ash
/bin/bsh
/bin/ksh
/usr/bin/ksh
/usr/bin/pdksh
/bin/tcsh
/bin/csh
/bin/test.sh
```

Utilización

Para que un usuario pueda utilizarlo hay que asignárselo en el archive /etc/passwd si el usuario ya existe en el sistema, como en este ejemplo el usuario admin.

```
[root@ns3 ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
.
```

```
.  
.  
apache:x:48:48:Apache:/var/www:/sbin/nologin  
squid:x:23:23::/var/spool/squid:/sbin/nologin  
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin  
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin  
ntp:x:38:38::/etc/ntp:/sbin/nologin  
named:x:25:25:Named:/var/named:/sbin/nologin  
avahi:x:70:70:Avahi daemon:/sbin/nologin  
avahi-autoipd:x:100:101:avahi-autoipd:/var/lib/avahi-autoipd:/sbin/nologin  
distcache:x:94:94:Distcache:/sbin/nologin  
web:x:500:501::/var/www/html/ftp:/bin/bash  
accesso_ftp:x:501:503::/home/acceso_ftp:/sbin/nologin  
hyadmin:x:502:504::/home/hyadmin:/bin/bash  
admin:x:503:0::/home/admin:/bin/test.sh
```

Si se quisiera crear un usuario y usar nuestro shell se debería ejecutar el siguiente comando:
adduser “nombre_de_cuenta” -s test.sh -p “contraseña de la cuenta”

Ejemplo de Shell (vía prg en C con procesos sincronizados por un semáforo)

```
[root@ns ~]# gcc shell_sem_uai.c -o shell_sem_uai
```

La anterior línea muestra como se compila el archivo con nombre “shell_sem_uai.c” y queda como programa ejecutable “shell_sem_uai”, simplemente una vez hecho se debe usar como se describe en las secciones anteriores. A continuación se muestra el código fuente:

```
[root@ns ~]# cat shell_sem_uai.c  
//como todo programa en C se agregan las librerías que contienen las funciones existentes.  
#include <stdio.h>  
#include <fcntl.h>  
#include <stdlib.h>  
#include <string.h>  
#include <signal.h>  
#include <unistd.h>  
#include <sys/wait.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <sys/ipc.h>  
#include <sys/sem.h>  
#include <semaphore.h>  
  
// Declaración de funciones  
void proceso_hijo(int);
```

```
void proceso_padre(int);
int inicia(int);
int borra_s(int);
void P(int);
void V(int);
int semcall(int, int);

// desarrollo del programa principal
int main(int argc, char *argv[])
{
    int mutex; // semaforo binario
    mutex=inicia(1); //se lo inicializa
    pid_t pid;
    while (1) //ciclo infinito para que vuelva a ejecutarse.
    if ( 0 == (pid=fork()) ) // creación de un proceso
    {
        proceso_hijo(mutex); // ejecución del código del proceso hijo
    }
    else
    {
        proceso_padre(mutex); // ejecución del código del proceso padre
        waitpid( pid, NULL, 0 ); //espera del que el proceso hijo termine
    }
    borra_s(mutex); //devuelve los recursos al sistema
    exit(0);
}

//desarrollo de funciones
void proceso_padre(critica)
{ /* ejecuta uno de los 3 comandos a ingresar por un usuario: ls ó ll ó find como el ejemplo
anterior solo que espera el path para listarlo */
P(critica); //restringe el recurso
printf("ingrese comandos ej: ls, ll, find\n");
char aux[16];
scanf("%s", aux);
printf("ingrese directorio\n");
char aux2[16];
scanf("%s", aux2);
if (strcmp("ls",aux)==0)
{
    strcpy(aux,"/bin/ls ");
    strcat(aux,aux2);
    system(aux);
}
else
```



```

    if (strcmp("ll",aux)==0)
    {
        strcpy(aux,"/bin/ls -la ");
        strcat(aux,aux2);
        system(aux);
    }
    else
        if (strcmp("find",aux)==0)
        {
            strcpy(aux,"/usr/bin/find ");
            strcat(aux,aux2);
            system(aux);
        }
    printf("aprete enter para continuar");
    getchar();
    getchar();
    V(critica); //libera el recurso
    wait(0); //espera hasta q lo liberen.
}

void proceso_hijo(critica)
{
    P(critica); //utiliza el recurso
    system("/usr/bin/clear"); //limpia la pantalla
    system("/bin/cat /tmp/uai.txt"); //muestra un banner, debajo se muestra *1.
    V(critica); //libera el recurso
    exit(0); //libera el proceso
}

int inicia (valor)
{
    int semval;
    int id;
    union semun {
        int val;
        struct semid_ds *buf;
        ushort *array;
    } arg;
    if ((id=semget(IPC_PRIVATE, 1, (IPC_CREAT|0666))) == -1) // *2
    { //la línea anterior crea un valor nuevo y único con permiso de RW para todos.
        perror("Error al crear el semaforo.");
        return(-1);
    }
    arg.val = valor;
    if (semctl(id, 0, SETVAL, arg) == -1)

```

```
{
    perror("Error al inicializar el semaforo.");
    return (-1); //error en inicializacion
}
return(id);
}
//fusión que devuelve el recurso del semáforo al sistema
int borra_s (semaforo)
{
    if ( semctl(semaforo, 0, IPC_RMID, 0) == -1) //2*
    {
        perror("Error al eliminar el semaforo.");
        return(-1);
    }
    return 0;
}
//restringe el uso del recurso
void P(semaforo)
{
    if ( semcall(semaforo, -1) == -1 )
        perror("Error en operacion.");
}

//libera el recurso
void V(semaforo)
{
    if ( semcall(semaforo, 1) == -1 )
        perror("Error en operacion.");
}

int semcall(semaforo, operacion)
{
    struct sembuf sb;
    sb.sem_num = 0;
    sb.sem_op = operacion;
    sb.sem_flg = 0;
    return (semop(semaforo, &sb, 1) ); //devuelve -1 si error 2*
}
```

*1.\ Banner

```
[root@ns ~]# cat uai.txt

      uuuu  uuuu      aaaaaa      iiii
      u u u u      a    a      i i
      u u u u      a  aa  a      i i
      u u u u      a a a a      i i
      u u u u      a  aa  a      i i
      u u u u      a    a      i i
      u u u u      a aaaa a      i i
      u u u u      a a a a      i i
      u  uu  u      a a a a      i i
      u    u      a a a a      i i
      uuuuuu      aaaa  aaaa      iiii

[root@ns ~]#
```

*2.\ Comunicación entre procesos (IPC)

En UNIX/Linux se define tres operaciones fundamentales sobre semáforos:

- semget: Crea o toma el control de un semáforo
- semctl: Operaciones de lectura y escritura del estado del semáforo. Destrucción del semáforo
- semop: Operaciones de incremento o decremento con bloqueo

Como el lenguaje C no tiene un tipo "semáforo" predefinido, si queremos usar semáforos tenemos que crearlos mediante una llamada al sistema (semget). Esta llamada permite crear un conjunto de semáforos, en lugar de uno solo. Las operaciones se realizan atómicamente sobre todo el conjunto; esto evita interbloqueos

Al crear un semáforo se nos devuelve un número identificador, que va a funcionar casi igual que los identificadores de fichero de las llamadas open, creat, etc. La función semget nos permite además "abrir" un semáforo que ya esté creado. Así, por ejemplo, si un proceso crea un semáforo, otros procesos pueden sincronizarse con aquél (con ciertas restricciones) disponiendo del semáforo con semget.

Para darle un valor inicial a un semáforo, se utiliza la función semctl.

En Unix/Linux no ofrece las funciones clásicas P y V o equivalentes, sino que dispone de una función general llamada semop que permite realizar una gama de operaciones que incluyen las P y V. semctl también se emplea para destruir un semáforo.

Operaciones de control sobre semáforos con la función semctl

semid es un identificador de semáforo (devuelto previamente por semget) y semnum, el semáforo del conjunto sobre el que quieren trabajar. cmd es la operación aplicada; a continuación puede aparecer un parámetro opcional según la operación definida por cmd.

Las operaciones que se ejecutan en el programa son:

- GETVAL: retorna el valor actual del semáforo.
- SETVAL: se modifica el valor del semáforo.
- IPC_RMID: destruye el semáforo.

GUÍA DE ABORDAJE BIBLIOGRÁFICO

Unidad 1 - Introducción, Definición Y Evolución De Los Sistemas Operativos

Unidad 2 - Algoritmos De Planificación De Procesos Y De Gestión De La Memoria

Unidad 3 - Gestión De La Memoria Y Sincronización De Procesos

Stallings, W. (2005). Sistemas operativos: aspectos internos y principios de diseño. 5ta.ed. Pearson Prentice Hall. Madrid.

Unidad 4 - Dispositivos Periféricos Y Técnicas De Gestión De Almacenamiento De Información

Tanenbaum, AS. (2003). Sistemas operativos modernos. 2a.ed. Pearson Prentice Hall. Distrito Federal. México.

Unidad 5 - Virtualización Y Computación En La Nube.

Rafaels, RJ. (2015). Cloud Computing: From Beginning to End. CreateSpace Independent Publishing Platform. Carolina Del Sur.

Unidad 6 – Compiladores

Alfonseca, M; De la Cruz M; Ortega, A; Pulido, E. (2006). Compiladores e intérpretes: teoría y práctica. Pearson Prentice Hall. Madrid.