

PREGUNTAS PARCIAL TEORICO

- 1) En paginación por demanda, la selección del marco abre en la que se colocara una pagina que genere un fallo de pagina es realizada por la MMU. **FALSO**
- 2) En segmentación, los valores de las direcciones base y limite de cada segmento que se colocan en la tabla de segmentos es determinado por: **MMU**
- 3) Una vez finalizada la ejecución de una SysCall el cambio a modo usuario es realizado mediante la generación de una Excepción (interrumpida por software). **FALSO**
- 4) La cola de listos (Ready Queue) es consultada/utilizada por el Short Term Scheduler para determinar que proceso admitir en memoria. **FALSO**
- 5) Un proceso existe desde que el programa es cargado en el espacio de direcciones de memoria de este. **FALSO**
- 6) Indique cuales de las siguientes actividades permite la interrupción por clock. **PODER PROTEGER LA EJECUCION DE INSTRUCCIÓN DE E/S – PODER PROTEGER QUE UN PROCESO NO SE APROPIE DE LA CPU – PROTEGER EL ESPACIO DE DIRECCIONES**
- 7) En Linux/unix la ejecución de la SysCall modificara el PID (Process ID) del proceso que la ejecuta. **FALSO**
- 8) Durante el Context Switch (cambio de contexto) el dispatcher cambia a modo usuario luego de realizar el salto a la próxima instrucción del proceso que obtiene la CPU. **FALSO**
- 9) Para planificar procesos interactivos son convenientes algoritmos apropiativos (Preemptive). **VERDADERO**
- 10) Las SysCalls que invoca un proceso durante su ciclo de vida se ejecutan en: **EL CONTEXTO DE DICHO PROCESO**
- 11) La PCB de un proceso que ejecuta una SysCall bloqueante, por ejemplo un write() a un archivo, será colocada en **LA COLA DE WAITING O BLOCKED** hasta que la ejecución de la SysCall termine.
- 12) En paginación el chequeo de protección de memoria mediante las tablas de paginas es realizado por el **HARDWARE (MMU)** el cual avisa al **KERNEL** de algún error por medio de un Trap (excepción).
- 13) En segmentación todos los segmentos de procesos deberán ser de igual tamaño. **FALSO**
- 14) La atención de un Page Fault (Fallo de página) se realiza en: **MODO KERNEL**
- 15) Cuando en paginación por demanda un proceso genera un fallo de página, dicho proceso, es puesto en estado de **WAITTING O BLOCKED** mientras el fallo se resuelve.
- 16) En la técnica de paginación por demanda se busca que la tasa de fallos de paginas sea 0 (cero). **FALSO**
- 17) En paginación el espacio de direcciones de un proceso en memoria principal (RAM) podrá estar ubicado de forma no continua, situación que no se produce en segmentación donde los segmentos deberán estar uno continuo del otro. **FALSO**
- 18) El tiempo consumido durante un cambio de contexto es considerado no productivo. **VERDADERO**
- 19) Indique cuales de los siguientes eventos ocurrirán cuando un proceso que se encontraba en estado de “Espera Swappeado”, cambia a estado “Listo Swappeado”: **CAMBIA EL ESTADO DE SU PBC – SE CAMBIA DE COLA SU PBC**
- 20) El acceso por parte de un proceso a una posición de memoria no autorizada es detectado por: **LA MMU**
- 21) ¿El SO necesita tiempo de CPU? **VERDADERO. (Porque es software)**
- 22) ¿Pueden convivir en un mismo SO la modalidad Batch y la Interactiva? **VERDADERO. (Pero no pueden ser las dos juntas)**
- 23) ¿Puede un sistema monousuario ser multitarea? **VERDADERO**
- 24) ¿Puede un sistema multiusuario ser monotarea? **VERDADERO**
- 25) ¿Los sistemas time sharing son una consecuencia de la multiprogramación? **VERADERO**
- 26) ¿Puede un programa ejecutarse desde el disco? **FALSO (desde la ram / debe cargarse en memoria)**
- 27) ¿Puedo planificar el uso de la CPU si no cuento con memoria secundaria? **FALSO. (Para planificar precisas swappear)**
- 28) La interrupción por clock impide que un proceso se apropie del procesador. **VERDADERO**
- 29) La interrupción es externa al proceso. **VERDADERO**
- 30) Un intento de acceder a una dirección ilegal se trata como una excepción o trap. **COMO TRAP**
- 31) Un proceso puede acceder al espacio de direcciones de otro proceso si esta en modo usuario. **FALSO (en modo usuario accede solo a su propio espacio) (debería ser modo supervisor sino)**
- 32) Una llamada al sistema (system call) genera aun proceso del sistema operativo. **FALSO**

- 33) La llamada al sistema es la forma que tiene la aplicación de comunicarse con el sistema operativo. **VERDADERO**
- 34) Si tengo varios procesos orientados a entrada/salida, las colas de solicitudes a los dispositivos estarán vacías. **FALSO**
- 35) ¿El SO permite al usuario abstraerse del hardware y su manejo? **VERDADERO**
- 36) ¿Es lo mismo el kernel que el SO? **FALSO**
- 37) La memoria principal es un recurso del tipo multiplexada en el espacio. **VERDADERO**
- 38) El procesador en un sistema monoprocesador es un recurso del tipo multiplexada en el tiempo a cada proceso. **VERDADERO**
- 39) ¿Open (archivo) se implementa como una system call? **VERDADERO**
- 40) ¿Date se implementa como una system call? **VERDADERO**
- 41) Un proceso tiene un stack en modo usuario y un stack en modo supervisor. Como no se usan a la vez, ocupan la misma dirección de memoria. **VERDADERO**
- 42) El estado del proceso esta en la PCB. **VERDADERO**
- 43) Un proceso crea a otro mediante una system call. **VERDADERO**
- 44) La cola de procesos esta en el disco. **FALSO (en memoria)**
- 45) Cuando un proceso se crea, esta en disco. **FALSO**
- 46) El proceso padre crea al hijo en su propio espacio de direcciones. **FALSO**
- 47) Las tablas de archivos que usa el proceso forman parte de su contexto.
- 48) La PCB se crea a partir que el proceso se carga en memoria. **FALSO**
- 49) El proceso padre y el hijo comparten la PCB. **FALSO**
- 50) Si no fuera por la E/S, los procesos no necesitarían system call. **FALSO**
- 51) Para que un proceso pueda acceder al espacio de direcciones de otro proceso, debe estar en modo supervisor. **VERDADERO**
- 52) El contexto de un proceso es lo mismo que su espacio de direcciones. **FALSO**
- 53) Para implementar prioridad dinámica o aging, se tiene en cuenta: **CUANTO TIEMPO DE ESPERA TIENE ACUMULADO.**
- 54) ¿Es lo mismo cambio de contexto que cambio de proceso? **VERDADERO**
- 55) ¿Es lo mismo cambio de contexto que cambio de modo? **FALSO**
- 56) Un cambio de contexto involucra un cambio de modo. **VERDADERO**
- 57) Un cambio de modo involucra un cambio de contexto. **FALSO**
- 58) Un fork exitoso produce cambios en la PCB del padre pues se almacena **INICIALIZA EL PUNTERO AL 1ER HIJO** del hijo.
- 59) El espacio de direcciones de un proceso está delimitado por los registros **BASE y LIMITE/LONGITUD.**
- 60) El fork devuelve dos valores: **0** al proceso hijo y **+0 o PID** al proceso padre.
- 61) El acceso no autorizado por parte de un proceso a una posición de memoria es detectado por: **EL HARDWARE.**
- 62) Las System Call se ejecutan en “Modo Privilegiado”. **VERDADERO**
- 63) Ante un cambio de contexto, indique cuales de estos elementos se guarda en la PCB: **ESTADO DEL PROCESO / PUNTERO DE LA PILA**
- 64) El chequeo de la existencia de una interrupción se realiza entre los pases de “Fetch” y “Execute” de cada ciclo de instrucción. **¿VERDADERA?**
- 65) El vector de interrupciones siempre debe estar en memoria. **VERDADERO**
- 66) Un system call fork, provocara cambio de contexto. **FALSO**
- 67) Un proceso swapeado en estado listo (ready to run) no compete por CPU. **VERDADERO**
- 68) El scheduler de short term se ejecuta con menos frecuencia que el de long term. **FALSO (corto plazo a largo plazo)**
- 69) El cambio de contexto lo hace el scheduler de long term. **FALSO**
- 70) Cuando un proceso se crea, se pone en la cola de procesos, hasta que es elegido por el scheduler de long term. **VERDADERO**
- 71) Cuando a un proceso se le termina su quantum, pasa a estado de espera. **FALSO**
- 72) El scheduler de medium term es quien decide el cambio entre nuevo y ready. **FALSO**
- 73) El scheduler de short term es quien hace pasar al proceso de estado ready a running. **FALSO**

- 74) En la planificación de CPU se trata de maximizar la productividad, minimizar el tiempo de respuesta. **VERDADERO**
- 75) El tiempo de retorno, es el tiempo desde que se inicia hasta que termina, sumando cpu, espera en colas. **VERDADERO**
- 76) Supongamos que un proceso esta en espera swapeado y se cumple el evento por el que estaba esperando. El proceso queda en estado de listo en memoria secundaria. **VERDADERO**
- 77) Según el diagrama visto: ¿puede un proceso pasar del estado de nuevo (creado) a listo swapeado? **SI**
- 78) Un proceso puede pasar de esperar en memoria secundaria a esperar en memoria real.
- 79) El scheduler de médium term maneja el grado de multiprogramación. **VERDADERO**
- 80) El disco permitió implementar la planificación de procesos. **VERDADERO**
- 81) Cuando a un proceso se le termina su quantum, pasa a estado de espera. **FALSA (Pasa a estado de preparados/listos)**
- 82) En un sistema monoprocesador, cuando se ejecuta la rutina de interrupciones, el resto de los procesos quedan en espera. **VERDADERO**
- 83) En un ambiente interactivo y batch, que se maneja con colas múltiples con multinivel, ¿conviene usar algoritmos apropiativos? **NO**
- 84) Indique cual es la combinación que representa la sucesión de actividades que realiza el dispatcher: **CAMBIO DE MODO – SALTO A PRIMER INSTRUCCIÓN A EJECUTAR**
- 85) Indique que puede ocurrir cuando tengo demasiados procesos orientados a I/O: **SE SATURAN LAS COLAS DE DISPOSITIVOS**
- 86) Cuando se carga un proceso en memoria, se hace en modo usuario. **FALSO**
- 87) En la administración particionada de memoria, no puede haber swapping. **VERDADERO**
- 88) En la administración particionada de memoria hay fragmentación interna. **VERDADERO**
- 89) En la estrategia best fit, se elige de entre las particiones disponibles, la que produzca menor fragmentación interna. **VERDADERA**
- 90) En la estrategia worst fit, se elige de entre las particiones disponibles, la que produzca mayor fragmentación interna.
- 91) La más rápida de las estrategias de asignación de particiones de memoria, es la first fit. **VERDADERA**
- 92) Resolver la dirección en el momento de la carga, exigirá que el proceso se ejecute siempre en el mismo lugar de la memoria.
- 93) Si se resuelven las direcciones en el momento de la compilación y quieren cambiarse, se debe recompilar.
- 94) En la administración particionada fija, el grado de multiprogramación lo da la cantidad de particiones. **VERDADERO**
- 95) La resolución de direcciones en el momento de la carga facilita la administración paginada. **VERDADERO**
- 96) ¿El kernel carga el proceso en memoria? **VERDADERO**