**CONDICIONES GENERALES:**

**Formato de entrega:**

* Electrónico individual en un único PDF conteniendo todas las respuestas a las preguntas y situaciones planteadas, incluyendo cualquier demostración o ejercicio. Una vez completado el documento se debe imprimir seleccionando impresora PDF.
* Para editar el nombre y documento deberá editar con Word la zona del encabezado de páginas
* No se aceptarán archivos separado formato Word / Excel / PDF, por lo que cualquier información deberá ser incorporada en el presente documento.
* Todas las entregas deberán ser realizadas a través de la plataforma Miel. Solo se puede subir un archivo de respuestas en formato PDF. La evaluación será devuelta por la misma plataforma.
* Esta evaluación de aprendizaje estará disponible para ser resuelta desde el **viernes 23 de octubre** hasta el **sábado 24 de octubre** luego de este plazo no se aceptarán más respuestas. Si Miel admite subir contenidos luego de ese plazo no serán evaluados.
* **IMPORTANTE:** No se pueden presentar el examen más de una vez, en caso de detectarse múltiples entregas se considerará no entregado.

**Forma de resolución de los ejercicios:**

* **Preguntas con opciones:** Se debe marcar la/s opción/es correctas con resaltado verde y negrita. Ej. **Opción Correcta**
* **Preguntas Verdadero/Falso:** se debe resaltar la opción correcta con verde y negrita. Ej. **Verdadero**

En caso de requerir justificación se deberá agregar inmediatamente a continuación de la opción correcta con letra Calibri (Cuerpo) o Arial de 11 sin ninguna negrita ni resaltado. En todos los casos las justificaciones no deberán tener más de 5 renglones.

* **Preguntas a desarrollar:** se deberán contestar inmediatamente debajo de la pregunta, no se deberá alterar la numeración del cuestionario enviado por la cátedra.

El texto deberá estar escrito con letra Calibri (Cuerpo) o Arial de 11 sin ninguna negrita ni resaltado. No se deberá superar la cantidad de renglones que indique la pregunta si es que así lo indica. No se permite el uso de esquemas numerados, pero sí viñetas tipo puntito.

* **Ejercicios prácticos:** Se deberán completar las grillas incluidas en la presente. Se podrán utilizar colores para resaltar las cuadriculas o situaciones que se deseen. Hacer aclaraciones con cuadros de texto siempre verificando que los mismos no tapen otros componentes de la evaluación.

El texto deberá estar escrito con letra Calibri (Cuerpo) o Arial de 11 sin ninguna negrita ni resaltado. No se deberá superar la cantidad de renglones que indique la pregunta si es que así lo indica. No se permite el uso de esquemas numerados, pero sí viñetas tipo puntito.

* **Preguntas de práctica de Laboratorio:** Se deberán responder utilizando texto, en caso de ser necesario incluir código el mismo deberá estar formateado con letra Lucida Console de 11 sin ninguna negrita ni resaltado. No se deberá superar la cantidad de renglones que indique la pregunta si es que así lo indica. No se permite el uso de esquemas numerados, pero sí viñetas tipo puntito.

**CONDICIONES DE APROBACIÓN:**

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Para nota de 4 a 6** | **Para nota 7 o más** |
| **Teoría** | 5 bien (B / B-)  1 regular más (R+) | 6 bien (B / B-)  2 regular más (R+) |
| **Práctica** | 1 bien (B / B-)  1 encarada (R / R-) | 1 bien (B / B-)  1 regular más (R+) |
| **Laboratorio** | 4 bien (B / B-)  1 regular más (R+) | 5 bien (B / B-)  1 regular más (R+) |

Las preguntas a resolver se evaluarán con las calificaciones B, B-, R+, R, R-, M siendo consideradas bien solo las dos primeras calificaciones.

Las respuestas de tipo múltiples opciones incompletas o donde se seleccionaron opciones erróneas serán consideradas mal sin importar si alguna de las incluidas es correcta o no.

La suma de dos o más preguntas R no implican un contenido bien resuelto, estas calificaciones son solo para que el alumno pueda identificar los puntos erróneos en su evaluación de aprendizaje y donde debe profundizar el proceso.

**CONTENIDO A RESOLVER TEÓRICO:**

**Marque Verdadero o Falso y justifique**

1. En un ambiente de monoprocesamiento y multiprogramación para una aplicación que solo realiza procesamiento (sin E/S) y cuyo procesamiento podría realizarse en varias tareas paralelas, resulta conveniente utilizar hilos a nivel de usuario (ULT)

**Verdadero**: Ya que el proceso no accede a E/S , los hilos tampoco solicitarían E/S, por lo tanto se bloqueara el proceso y podrán continuar con sus tareas en formas paralela

**Falso: no tengo multiprocesamiento ni entrada salida, entonces no conseguiría ningún beneficio, porq pierdo tiempo en el cambio de contexto**

* 1. Falso:

1. Una de las ventajas del mecanismo de IPC Shared Memory, es que los procesos que lo usan no requieren estar sincronizados, ya que la porción de memoria que se comparte está bajo la gestión y supervisión del Sistema Operativo
   1. Verdadero:

**Falso**: Los procesos son los que implementan las herramientas de sincronización para evitar conflictos en áreas criticas

**Falso:el acceso a la memoria compartira tiene que estar sicnronizado, tinene que usar mecanismo de sincro poer no se autogestiona**

1. En un momento dado y por un lapso de tiempo x, un sistema cuyo valor de saturación ρ >1 entonces el planificador de largo plazo se ejecuta más veces que el de corto plazo
   1. Verdadero:

**Falso**: El planificador corto plazo siempre se ejecutará más veces que el de largo plazo porque estará constantemente decidiendo que procesos toma la CPU, sin importar que los procesos continúen acumulándose en la cola de listos

**Falso: q el nivel de saturación sea mayor a 1, lo q significa es que están llegando procesos nuevos al sistema con una tasa superior a la que salen los procesos**

1. Cuando cada proceso de un conjunto espera por un mensaje de otro miembro del mismo conjunto, y no existe un mensaje en tránsito, entonces ocurre un Deadlock.

**Verdadero**: Los procesos quedaran esperando un mensaje que nunca va a llegar, por lo tanto, ninguno podrá seguir con su ejecución

* 1. Falso:

**Verdadero: si tengo varios procesos que stan bloqueados por la llegada de un mensaje de otro procesos de eese conjunot y no hay ninugn mensaje en curso voy a tener un deadlock**

**Responda** (en no más de 5 renglones)

1. ¿Indique que diferencias hay entre los mecanismos de IPC Pipe y Fifo?

El mecanismo de pipe se utiliza cuando los procesos tienen un parentesco, es un mecanismo unidireccional y que solo puede comunicarse localmente. En cambio, el fifo no es necesario que los procesos tengan un antepasado común, la comunicación es semidúplex y pude hacer entre computadoras diferentes.

**El pipe es unidireccional y para procesos emparentados, no esta representado por un archivo en el file system**

1. Un estudiante con intenciones de aprender a trabajar con threads se propuso realizar el típico programa “*Hola Mundo*”, pero de tal manera que cada palabra de esa frase sea impresa por un thread distinto. Entonces escribió un programa cuyo pseudocódigo es:

**Código principal Código Hilo 1 Código Hilo 2**

main() { { {

crear\_hilo 1; print “Hola “; print “Mundo “;

crear\_hilo 2; } }

exit();

}

Al ejecutarlo, obtuvo el resultado esperado por pantalla.

Sin embargo, notó algo curioso. Ejecutó el proceso 10 veces, y 2 de esas veces apareció en pantalla “*Mundo Hola*“.

Repitió la prueba, ejecutando el proceso otras 10 veces, y esta vez paso algo más extraño aún. Apareció 7 veces “*Hola Mundo*”, 1 vez “*Mundo Hola*” y 2 veces no apareció nada.

Se pide:

1. Explique a qué se debe cada uno de estos comportamiento y qué nombre recibe en la jerga

Se debe al orden en el que son ejecutados los procesos. Sin un mecanismo de sincronización, la manera en la que se ejecutaran es inpredecible. El nombre que recibe es Race Condition

1. Indique qué debería hacer el estudiante para que siempre pueda obtener la salida esperada.

Utilizar semáforos para sincronizar la ejecución de los 2 hilos

1. Proponga una modificación al código para que el proceso siempre funcione bien tomando en cuenta que es obligatorio que se mantenga la ejecución en threads propuesta (no se requiere codificación solo la propuesta)

Podría utilizar 3 semáforos, A y B , A inicializado en 1 y B inicializado en 2, entonces el hilo 1 pide el semáforo, muestra hola y habilitad B, luego el hilo 2 pide por B y ejecuta . Finalmente el tercer semáforo ubicado previo al exit que se habilite al finalizar el hilo 2 para garantizar que siempre se ejecuten antes de que el procesos principal finalice

**a-Los hilos no están sincronizados, Race condition, si se ejecuta el proceso principal antes q los hilos no se imprime lo de los hilos.**

**b-Se debe implementar sincro, para que se impriman bien y que el padre espere a los hilos**

**Seleccione la/las opciones correctas y justifique el motivo por el que se selecciona o no**

1. El Process Control Block y los estados de los procesos
   1. El PCB de un proceso solo se modifica cuando el proceso pasa del estado de listo a ejecución y viceversa

El PCB va cambiando constantemente ya que almacena la dirección de la próximo instrucción, por lo tanto se va actualizando luego de cada ejecución.

**El pcb se modifica con cualquier cambio de estado del proceso**

* 1. **El PCB de un proceso es creado por el S.O. cuando se crea el proceso y solo puede ser accedido por dicho proceso**

El SO crea el PCB para tener la información necesaria de dicho proceso, cada procesos tendrá su propio PCB. El SO necesita acceder a el para conocer el estado de cada proceso

**El proceso no accede al pcb , el único que accede al pcb es el so**

* 1. Un proceso en estado de ejecución solo puede pasar al estado bloqueado o listo

Puede pasar a un estado de ejecución a finalizado o a suspendido o abandonado

**Puede pasar a finalizado**

* 1. **Los cambios de estado de un proceso son siempre generados por la ocurrencia de una interrupción**

Para que se haga un cambio de estado en el proceso necesito una interrupción por software(Syscall) o una interrupción por hardware, esto generara que se pase de un estado a otro dependiendo de la rutina de atención de la interrupción

**Bien**

* 1. Nunca puede haber más de un proceso en estado de ejecución

Si tengo varios procesadores puedo tener más de un proceso en ejecución. Si tengo un solo procesador debo desalojar al proceso actual para para comenzar a ejecutar el otro

**Verdadero solo en entorno de multiprocesaiento**

1. Se dispone de un sistema monoprocesador y multitarea, con un algoritmo de planificación non-preemptive (sin desalojo). En dicho sistema se tiene el proceso A, en Running, el proceso B en Blocked y el proceso C en Ready:
   1. Si el proceso A realizara un Syscall, se ejecutará el proceso nulo, hasta tanto dicho Syscall se satisfaga.

Mientras se satisface la rutina del Syscall del proceso A, se pasa uno de los procesos restantes a ejecución, como B esta bloqueado y solo C esta ready , se comienza a ejecutar C

**Falso: porque el proceso c esta listo**

* 1. **Si llegara un nuevo proceso al sistema, A permanecería en Running de todos modos.**

Una vez que llegue un nuevo proceso , los pasos a seguir a continuación dependerán del algoritmo de planificación utilizado

**Verdadero : la llegada de un nuevo proceso, genera una interrupción se hace un hard switch, se pone en listo y luego se sigue ejecuando el proceso**

* 1. Si llegara una interrupción de hardware, la RAI se atenderá recién luego de que A libere la CPU por sus propios medios.

La interrupción tiene mayor prioridad que el procesos que se esté ejecutando, por lo tanto el proceso A se interrumpirá, se atenderá la RAI y luego se terminara la ejecución de dicho proceso

**Falso: las interrupciones igual se atienden**

* 1. Si el proceso B ya está en condiciones de volver a Ready, tendrá que esperar a que A decida abandonar el procesador, ya que la transición Blocked a Ready requiere la intervención del S.O. y eso no sería posible ya que el procesador está ocupado con A.

**Falso:la finalización del motivo del bloqueo del procesos b se manifiesta a través de una interrupción uqe se atiene y luego se continuado con la ej del proceso a**

Cuando B este en condición de volver a Ready se genera una IHE , por lo tanto el SO atenderá la interrupción, pasara a B de Blocked a Ready y lo que pase a continuación dependerá del algoritmo de planificación utilizado

* 1. Si el proceso A hace una llamada al syscall V(S), el S.O. lo pasará al estado Blocked y le otorgará el procesador al proceso C.

A no estaría bloqueado, sino que habilitaría el semáforo

**No pasa al estado bloqueado, hace syscal, se eqjecuta la primitiva y se sigue ejecuando a**

1. La finalización de una rutina de pedir un semáforo en Linux provoca una:
   1. Interrupción de SW

El proceso es quien genera genera una señal de interrupción, por lo tanto se considera como un Syscall

* 1. Interrupción de hw externa
  2. Trap
  3. Ninguna de las anteriores

No hay ningún tipo de interrupción, solo finaliza la rutina

**CONTENIDO A RESOLVER PRÁCTICO:**

1. Un determinado gimnasio recibe el ingreso de dos tipos de personas, deportistas extremos y amateur. Solo puede haber un máximo de 10 personas dentro del sitio teniendo en cuenta que para el ingreso se permite pasar a una persona amateur cada 2 deportistas extremos. Una vez ingresados, todos se dirigen a la sala de elongación con capacidad para 4 personas bajo la restricción de que nunca puede haber más 3 deportistas extremos o 2 deportistas amateur. Una vez finalizado la etapa de calentamiento, se dirigen a realizar ejercicios de musculación. En esta habitación hay espacio para 10 personas, existen 18 pesas de las cuales requieren 2 cada deportista profesional y 1 cada deportista amateur. Finalizada su rutina, se dirigen a retirarse del lugar. Los deportistas liberan la sala de fuerza y se retiran solos del gimnasio. Sin embargo, los deportistas extremos finalizada la rutina de fuerza, liberan la sala pero aguardan en una sala de espera con capacidad para 2 personas, en espera de otro deportista extremo. Una vez aparejados, cada uno notifica mediante un mensaje al único proceso supervisor de su retiro y el mismos le responde otro mensaje autorizando su salida del establecimiento

Se solicita generar la sincronización correspondiente indicando los procesos intervinientes, los recursos involucrados haciendo uso de las primitivas P(), V(), send(), receive() y funciones en pseudocódigo que indiquen las acciones que se están realizando en cada paso.

En el caso de utilizar mensajes se deberá indicar si las instrucciones de comunicación son bloqueantes o no

Si pido 2 recursos seguidos hay riesgo de deadlock

Acá como tengo 10 personas y tengo 18 mancuenrnas no es necesario un mutex ya que los deportistas van a poder tomar las mancuernas necesarias

MP(máximo de personas)=10 DA(deportista amateur)=0 SE(sala elongación)= 4

ED(elongación deportista)=3 EA(elongación amateur)=2 SM(sala musculación)=10

LP(levantamiento pesas)=18 SRD(retiro sala deportista)=2

Deportista Extremo Deportista Amateur

P(MP) P(DA)

V(DA) P(DA)

P(SE) P(MP)

P(ED) P(EA)

elongar(deportistaExtremo) P(SE)

V(SE) elongar(deportistaAmateur)

V(ED) V(SE)

P(SM) V(EA)

P(LP) P(SM)

P(LP) P(LP)

levantarPesas(deportistaExtremo) levantarPesas(deportistaAmateur)

V(LP) V(LP)

V(LP) P(SM)

V(SM) retirarse(deportistaAmateur)

P(SRD) V(MP)

Send()

Recieve()

Recieve()

V(SRD)

retirarse(deportistaExtremo)

P(A)

V(MP)

Retirarse de Gimnasio

Recieve()

Recieve()

Send()

Send()

Deportistas extremos

Los Send() no son necesario que sean bloqueantes ya que necesito recibir luego la confirmación a través de otro mensaje

Los Revieve() es necesario que sean bloqueantes para así sincronizar la retirada de los 2 deportistas extremos, cuando ambos hayan notificada que están listos para retirarse y esa notificación sea confirmada, estarán en condiciones de seguir su ejecución

Retirada de Gimnasio

Los Recieve() bloqueantes para que espere los mensajes de los 2 deportistas que requieren retirarse .Respecto a los Send(), no requieren que sean bloqueantes

1. Un sistema tiene dos canales de entrada / salida, en uno está conectado el grabador de cintas y en el otro el disco rígido.

El sistema usa un algoritmo de planificación Round Robin con prioridad fija, con un quantum máximo de 15u, con desalojo del procesador.

En un determinado momento llega un proceso con prioridad igual a 7, que realiza lo siguiente:

* **Proceso A**: ejecuta 5u, lee de disco 25u, ejecuta 15u, escribe en cinta 20u, ejecuta 25u, y finaliza

A las 30u llega un segundo proceso con prioridad igual a 5, que realiza lo siguiente:

* **Proceso B**: ejecuta 5u, lee de disco 20u, ejecuta 10u, escribe en cinta 30u, ejecuta 10u y finaliza.

**Se pide:** Realizar el diagrama temporal de procesamiento, indicando claramente los tiempos de ejecución de los procesos, de las rutinas del sistema operativo, las interrupciones que se producen para posibilitar cada cambio (con su clasificación), los conflictos (en caso de existir) y cuál fue el criterio utilizado para su resolución, evaluación de prioridades y su resolución.

**Operaciones del Planificador** (las instrucciones 2 a 5 son atómicas)**:**

1- Propuestos - Listos: 10u

2- Listos - Ejecutando: 10u

3- Ejecutando - Listos: 5u

4- Ejecutando - Bloqueados E / S: 5u

5- Bloqueados E / S - Listos: 5u

6- Ejecutando - Terminados: 15u

**Grilla para la resolución (**Complete debajo de cada grilla todo lo indicado en la sección “se pide”)**:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.O. A | 1 | 1 | 2 | 2 |  | 4 |  |  |  |  |  | 5 |  | 2 | 2 |  |  |  |  | 4 |  |  |  |  | 5 |  | 2 | 2 |
| S.O. B |  |  |  |  |  |  | 1 | 1 | 2 | 2 |  |  | 4 |  |  |  |  | 5 |  |  | 2 | 2 |  |  |  | 4 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prio A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prio B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **5** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** | **65** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | **105** | **110** | **115** | **120** | **125** | **130** | **135** | **140** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Cinta |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| | IHE |IHE |IHE |IHE(clock) |IHE

| Syscall |Syscall |Syscall |Syscall

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.O. A |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  | 2 | 2 | 6 | 6 | 6 |  |  |  |  |  |  |  |  |  |
| S.O. B |  |  |  |  | 5 |  |  | 2 | 2 |  |  | 6 | 6 | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prio A |  |  |  |  |  | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prio B |  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **145** | **150** | **155** | **160** | **165** | **170** | **175** | **180** | **185** | **190** | **195** | **200** | **205** | **210** | **215** | **220** | **225** | **230** | **235** | **240** | **245** | **250** | **255** | **260** | **265** | **270** | **275** | **280** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HD |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Cinta |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| |IHE |Syscall |

|IHE(clock) | Syscall

Al momento en que llega el nuevo proceso (30 u) como no hay procesos que puedan hacer un Syscall , no queda otra opción que se produzca por una interrupción de hardware externo

A los 55u recibo una IHE y un Syscall, atiendo primero la IHE (pasar de bloquea a listo el procesos A) y luego el Syscall(bloquear procesos B) debido a que la interrupción por hardware externo tiene mayor prioridad que un Syscall

Al momento 85u se produce una IHE que me indica que finalizo la lectura del disco, interrumpo la ejecución del proceso A, hago un context switch, por lo cual no lo desalojo el proceso A del procesador, atiendo la interrupción (desbloquear proceso B) Y continuo el quantum del proceso A.

En 95u no evalúo las prioridades de los procesos ya que, una vez que termina el quantum de A llega, al mismo tiempo, un Syscall que provoca que el proceso A pase a bloqueado por E/S, por lo tanto, solo queda el proceso B para pasar a estado de ejecución

En 125u el proceso A finaliza la escritura en cinta y se produce un IHE, el proceso B hace un Syscall para empezar escribir sobre la cinta, atiendo primero la interrupción por hardware ya que es mas prioritaria que el Syscall

En 155u el quantum del Proceso A finaliza, pero como no tengo otro proceso en la cola de listo vuelvo a darle otro quantum para que se siga ejecutando A

En 160u el proceso B finaliza la escritura en cinta, lanza un IHE , atiendo la interrupción sin sacar a A del procesador(context switch) y continuo con el quantum de A.

En 170 , el proceso A realizar un Syscall para ejecutar le rutina de finalización pero esta vez tengo ambos procesos aptos para ser ejecutados por lo cual sí es necesario analizar las prioridades. Como B tiene una prioridad mayor que A , es el elegido para ser el próximo en pasar a ejecución

La llegada de un proceso nuevo produce un syscall

**CONTENIDO A RESOLVER DE PRÁCTICA DE LABORATORIO:**

1. Dado el siguiente código en Bash:

#/bin/bash

IFS=".,."

funcion(){

a= cat $1

for b in a

do

echo "$b ///"

done

}

if [[ $# < 2 ]]

then

echo "La cantidad de parametros recibidos es incorrecta" #completar

exit

fi

if test -r $1

then

echo "El archivo no tiene permisos de lectura " #completar

exit

fi

funcion $1

* 1. ¿Cuál es el objetivo del Script? No se requiere explicar el paso a paso.

El objetivo del script es listar los campos de un archivo utilizando como “,” como separador

Objetivo: dividir el contenido de un archivo en base al separador ";”

* 1. Completar los espacios indicados con “…”.
  2. ¿Encuentra errores en el Script? De ser así, indíquelos. No se considerarán errores espacios faltantes o de sobra.

El IFS=”.,.” debe ser cambiado por IFS=”,”

Dentro de la función a= cat $1 debería estar rodeado entre $() -> a=$(cat $1)

El número de parámetro por el cual se pregunta podría ser 1, ya no se usa en ningún momento $2

En el if test -r $1 debe ser if ! [ [ -r $1 ] ]

1. ¿Se le ocurre otra forma de lograr la misma funcionalidad que el script del punto 12? De ser así descríbala.

IFS=”,”

if [[ $# < 1 ]]

then

echo “La cantidad de parámetros recibidos es incorrecta

exit

fi

if ! [[ -r $1 ]]

then

echo “El archivo no tiene permiso de lectura”

exit

fi

while IFS= read -r line

do

for i in $line

do

echo “$i ///”

done

done < $1

Cambio el separador a “,” leo una por una las líneas del archivo y las guardo en la variable $line, usando como separador “,” , luego recorro con un foreach esa variable line y muestro por pantalla los campos de la línea leída

1. ¿Qué son las señales y para qué se pueden utilizar en un script bash?

Las señales son notificaciones asincrónicas que se envían a los procesos para informarle un evento. La llegada de una señal modifica la ejecución normal del proceso. Las señales pueden utilizarse para para terminar un proceso, o pueden ser modificadas para ejecutar una acción determinada al momento de ser recibidas

Mecanismos para que los procesos se puedan comunicar entre si ,

Con relación a los eventos en Powershell seleccione la/s opción/es correcta/s:

* 1. Es como se denomina a las señales en Powershell
  2. Son notificaciones de la ocurrencia de una acción disparadas por el sistema operativo
  3. **El código asociado al evento permite utilizar las variables declaradas como locales en el script que ejecuta Register-ObjectEvent** ->Voy a perder el contenido de esa variable, solo lo voy a poder usar en el contexto propio del evento
  4. Solo se pueden utilizar variables globales en el código asociado a ellos

Falso, puedo usarlo dentro de un evento

* 1. Más de una opción es correcta (indique cuáles)
  2. Ninguna opción es correcta->opcion correcta

1. Explicar cómo funciona y procesa texto el comando awk

AWK interpreta cada línea como uno o varios campos separados por un espacio o tab. La entrada y/o la salida pueden ser el estándar o puede especificarse un archivo determinado, de la misma manera con la salida. El código de AWK puede ser escrito en otro archivo y llamarlo cuando se ejecuta la línea de AWK

1. En base al siguiente script Powershell, responder:

function Get-Something() {

param(

[Parameter( ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true) ]

[switch]$value

)

Begin { $count = @{ } }

Process {

$someValue = $value.Split(" \_ ")[1]

foreach ($x in $someValue) {

$count[$x]++

}

}

End {

Write-Output "Resultado Final:"

foreach ($x in $count) {

Write-Output $x

}

}

}

* 1. ¿Cuál es el objetivo del script? Explicar el resultado obtenido, no el paso a paso

Lista la cantidad de veces que aparece una clave numérica en una cadena

Cuanta la cantidad de

* 1. ¿Qué errores existen en el código?

Evitar declarar a $value como tipo switch ; En la línea del Split sacar el [1], esto provoca que solo me devuelva el número del segundo campo

* 1. ¿Cuál es el resultado que se obtiene por pantalla?

El contenido del hashmap ordenado en 2 columnas, una de claves numéricas y otra de valores con la cantidad de ocurrencias

1. Tomando en cuenta el script del punto 17 ¿Cómo se debe ejecutar la función para que el script devuelva un resultado válido?

Para que de un resultado valido podría pasar como parámetro el hashmap $count dentro de ciclo, lo que generaría que las ocurrencias se sumen sobre ese hashmap, por lo tanto el Write-host de un último ciclo daría las ocurrencias totales de cada elemento

Ejecutarlo con un pipe-> get content nombre de l archivo y un pipe