1. **Algoritmos de planificación de sistemas operativos.**

**1.1. Generalidades.**

Cuando un PC es multiprograma, se suele dar el caso de que varios procesos compiten por la CPU al mismo tiempo. Si solo existe una CPU disponible se tiene que decidir cuál de los procesos listos para ejecutarse será ejecutado a continuación. De esto se encarga una parte del S.O. que se llama planificador de procesos y el algoritmo que utiliza se llama algoritmo de planificación.

**1.2.Categorías de algoritmos de planificación.**

Distintos entornos necesitan diferentes algoritmos de planificación y los diferentes S.O. tienen sus propios objetivos por lo que el planificador de procesos tiene que optimizar no será lo mismo en todos los sistemas. Tres de los entornos más destacables son:

* Procesamiento por lotes
* Interactivo
* Tiempo real.

**1.3.Metas de los algoritmos de planificación**

Para poder diseñar un algoritmo de programación, es necesario tener idea de lo que debe hacer un algoritmo. Algunos objetivos dependen del entorno como el procesamiento por lotes, interactivo o de tiempo real), pero hay también algunos otros que son deseables en todos los casos.

**1.3.1. Todos los sistemas**

* Equidad – Otorgar a cada proceso una parte justa de la CPU
* Aplicación de políticas – Verificar que se lleven a cabo las políticas establecidas
* Balance – Mantener ocupadas todas las partes del sistema.

**1.3.2. Sistemas de procesamiento por lotes**

* Rendimiento – Maximizar el número de trabajos por hora
* Tiempo de retorno – Minimizar el tiempo entre la entrega y la terminación
* Utilización de la CPU – Mantener ocupada la CPU todo el tiempo

**1.3.3. Sistemas interactivos**

* Tiempo de respuesta – Responder a las peticiones con rapidez
* Proporcionalidad – Cumplir las expectativas de los usuarios

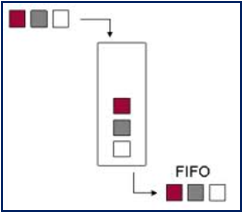
**1.3.4. Sistemas de tiempo real**

* Cumplir con los plazos – Evitar perder datos
* Predictibilidad – Evitar la degradación de la calidad en los sistemas multimedia

**1.4.Algoritmos de Planificación en sistemas de procesamiento por lotes.**

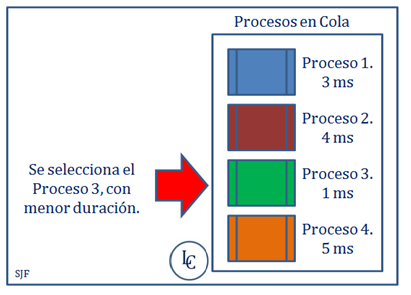
**1.4.1. FIFO:**

Acrónimo de “First in, first out” (primero que entra, primero que sale). Con este algoritmo la CPU se asigna a los procesos en el orden en el que la solicitan. Solo hay una sola cola de procesos listos. Cuando el primer trabajo entra al sistema, se inicia de inmediato y se le permite ejecutarse. A medida que van entrando otros trabajos, se colocan al final de la cola. Si el proceso en ejecución se bloquea, el primer proceso en la cola se ejecuta a continuación. Cuando un proceso bloqueado pasa al estado listo, al igual que un trabajo recién llegado, se coloca al final de la cola. Este algoritmo es fácil de comprender e igualmente sencillo de programar.



**1.4.2. SJF:**

Acrónimo de Shortest Job First (trabajo más corto primero) y algoritmo que supone que los tiempos de ejecución se conocen de antemano. Cuando hay varios trabajos de igual importancia esperando a ser iniciados en la cola de entrada, el planificador selecciona el trabajo más corto primero.



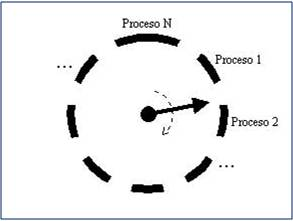
**1.4.3.SRTN:**

Shortest Remaining Time Next (menor tiempo restante a continuación). Algoritmo donde el planificador siempre selecciona el proceso cuyo tiempo restante de ejecución sea el más corto. De nuevo, se debe conocer el tiempo de ejecución de antemano. Cuando llega un nuevo trabajo, su tiempo total se compara con el tiempo restante del proceso actual. Si el nuevo trabajo necesita menos tiempo para terminar que el proceso actual, éste se suspende y el nuevo trabajo se inicia.

**1.5.Algoritmos de Planificación en sistemas interactivos.**

**1.5.1. Round Robin:**

Uno de los algoritmos más antiguos, simples, equitativos y de mayor uso es el de turno circular (round-robin). A cada proceso se le asigna un intervalo de tiempo, conocido como quántum, durante el cual se le permite ejecutarse. Si el proceso se sigue ejecutando al final del quántum, la CPU es apropiada para dársela a otro proceso. Si el proceso se bloquea o termina antes de que haya transcurrido el quántum, la conmutación de la CPU se realiza cuando el proceso se bloquea. Un quántum con un valor entre 20 y 50 mseg es lo más adecuado para un uso óptimo de la CPU por proceso.



**1.5.2. Por prioridad:**

La idea básica es simple: a cada proceso se le asigna una prioridad y el proceso ejecutable con la prioridad más alta es el que se puede ejecutar. Para evitar que los procesos con alta prioridad se ejecuten de manera indefinida, el planificador puede reducir la prioridad del proceso actual en ejecución en cada pulso del reloj (es decir, en cada interrupción del reloj). Si esta acción hace que su prioridad se reduzca a un valor menor que la del proceso con la siguiente prioridad más alta, ocurre una conmutación de procesos. De manera alternativa, a cada proceso se le puede asignar un quántum de tiempo máximo que tiene permitido ejecutarse.

**1.5.3. Planificación garantizada:**

Si hay por ejemplo 5 usuarios conectados mientras está trabajando, recibirá aproximadamente 1/5 del poder de la CPU. De manera similar, en un sistema de un solo usuario con 5 procesos en ejecución, mientras no haya diferencias, cada usuario debe obtener 1/5 de los ciclos de la CPU.

Para cumplir esta promesa, el sistema debe llevar la cuenta de cuánta potencia de CPU ha tenido cada proceso desde su creación. Después calcula cuánto poder de la CPU debe asignarse a cada proceso al saber el tiempo desde que se creó dividido entre 5 en este caso.

**1.5.4. Planificación por partes equitativas:**

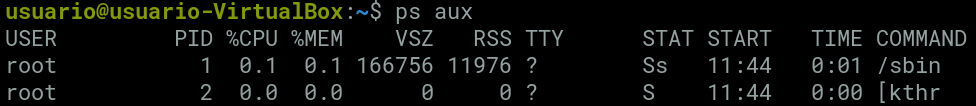
Hasta ahora hemos asumido que cada proceso se planifica por su cuenta, sin importar quién sea su propietario. Como resultado, si el usuario 1 inicia 9 procesos y el usuario 2 inicia 1 proceso, con la planificación por turno circular o por prioridades iguales, el usuario 1 obtendrá 90 por ciento del tiempo de la CPU y el usuario 2 sólo recibirá 10 por ciento. Para evitar esta situación, algunos sistemas toman en consideración quién es el propietario de un proceso antes de planificarlo. En este modelo, a cada usuario se le asigna cierta fracción de la CPU y el planificador selecciona procesos de tal forma que se cumpla con este modelo. Por ende, si a dos usuarios se les prometió 50 por ciento del tiempo de la CPU para cada uno, eso es lo que obtendrán sin importar cuántos procesos tengan en existencia.

**1.6.Algoritmos de Planificación en sistemas de tiempo real:**

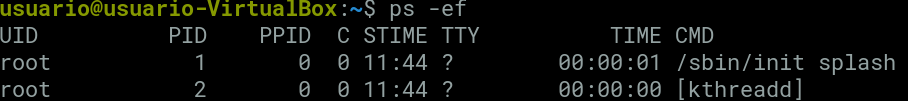
En un sistema de tiempo real, el tiempo desempeña un papel esencial. Por lo general, uno o más dispositivos físicos externos a la computadora generan estímulo y la computadora debe reaccionar de manera apropiada a ellos dentro de cierta cantidad fija de tiempo. En general, los sistemas de tiempo real se categorizan como de tiempo real duro, lo cual significa que hay tiempos límite absolutos que se deben cumplir, y como de tiempo real suave, lo cual significa que no es conveniente fallar en un tiempo límite en ocasiones, pero sin embargo es tolerable. En ambos casos, el comportamiento en tiempo real se logra dividiendo el programa en varios procesos, donde el comportamiento de cada uno de éstos es predecible y se conoce de antemano. Por lo general, estos procesos tienen tiempos de vida cortos y pueden ejecutarse hasta completarse en mucho menos de 1 segundo. Cuando se detecta un evento externo, es responsabilidad del planificador planificar los procesos de tal forma que se cumpla con todos los tiempos límite.

1. **Opciones de interés del comando ps.**

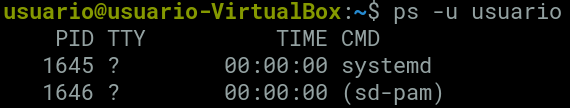
* ps aux: Enseña todos los procesos que se están ejecutando, incluyendo información detallada sobre el usuario, el estado del proceso y el uso de recursos.



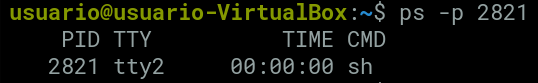
* ps -ef: Enseña todos los procesos en ejecución en el sistema, incluyendo información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.



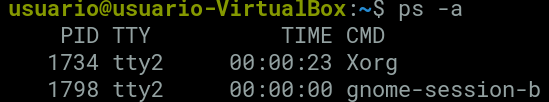
* ps -u [usuario]: Enseña todos los procesos en ejecución para un usuario específico. Reemplaza [usuario] con el nombre de usuario deseado.



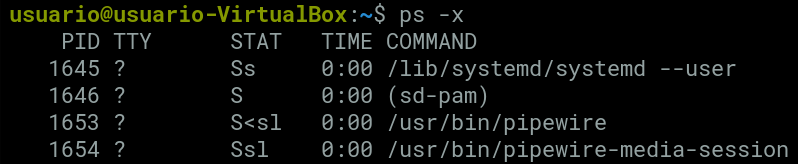
* ps -p [pid]: Enseña información detallada sobre un proceso específico, identificado por su identificador de proceso (PID). Reemplaza [pid] con el PID deseado.



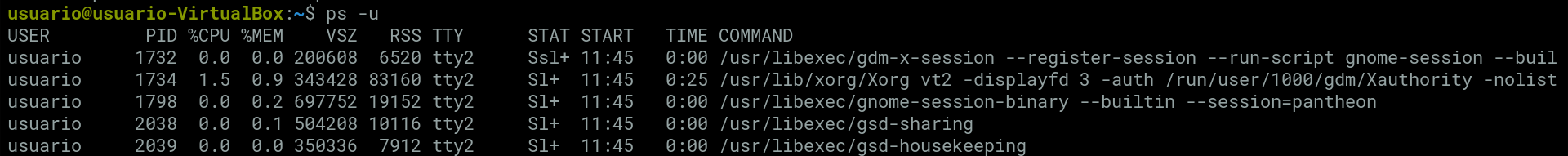
* ps -a: Muestra todos los procesos, incluyendo los que no tienen un terminal asociado.



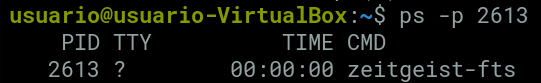
* ps -x: Muestra todos los procesos, incluyendo los que no tienen un terminal asociado y los procesos de sistema.



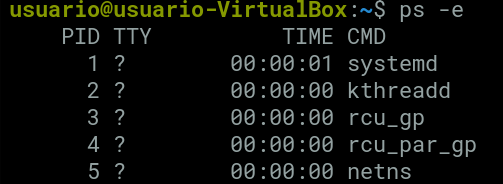
* ps -u: Muestra información detallada sobre el usuario, el estado del proceso y el uso de recursos.



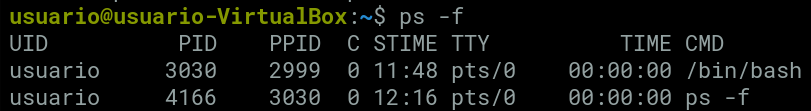
* ps -p: Muestra información detallada sobre un proceso específico, identificado por su PID.



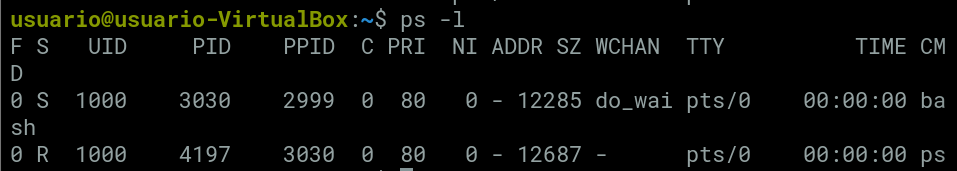
* ps -e: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.



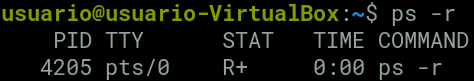
* ps -f: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos, el comando completo utilizado para iniciar el proceso y los procesos hijos del proceso.

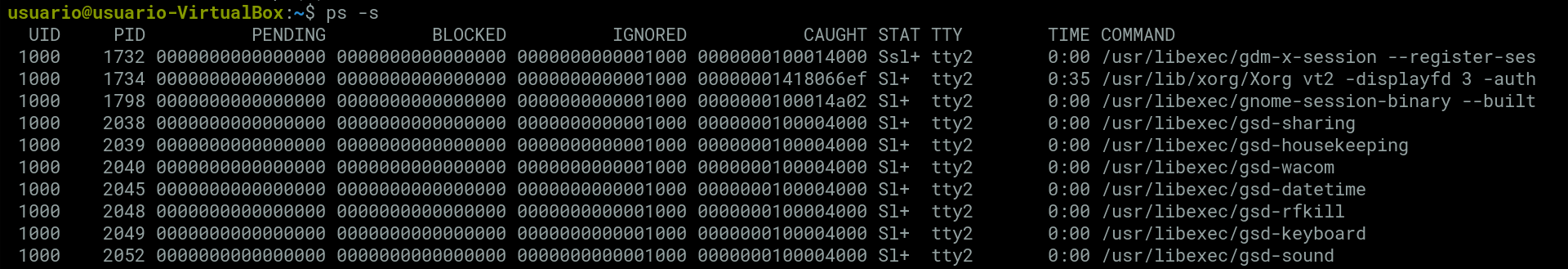


* ps -l: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos, el comando completo utilizado para iniciar el proceso y los procesos hijos del proceso.



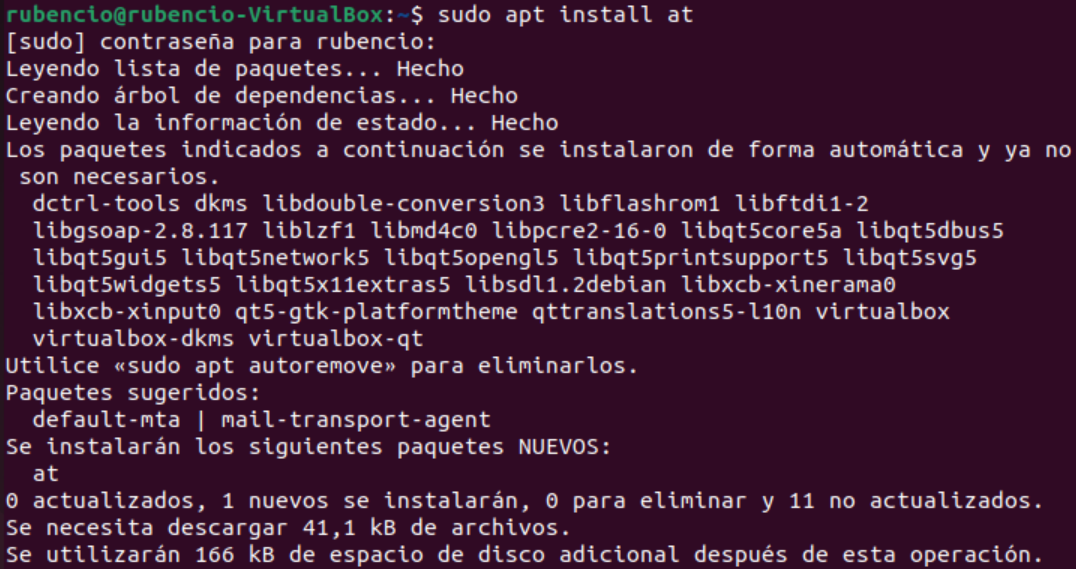
* ps -r: Muestra solo los procesos en ejecución.



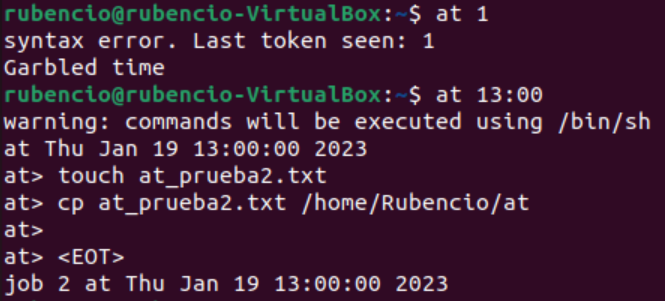
* ps -s: Muestra información detallada sobre el usuario, el estado del proceso, el uso de recursos y el comando completo utilizado para iniciar el proceso.

1. **Opciones de interés del comando at.**

El comando at en Linux se utiliza principalmente para la programación única de tareas. Si este comando no se encuentra en el sistema, se podrá obtener mediante el comando sudo apt install at.

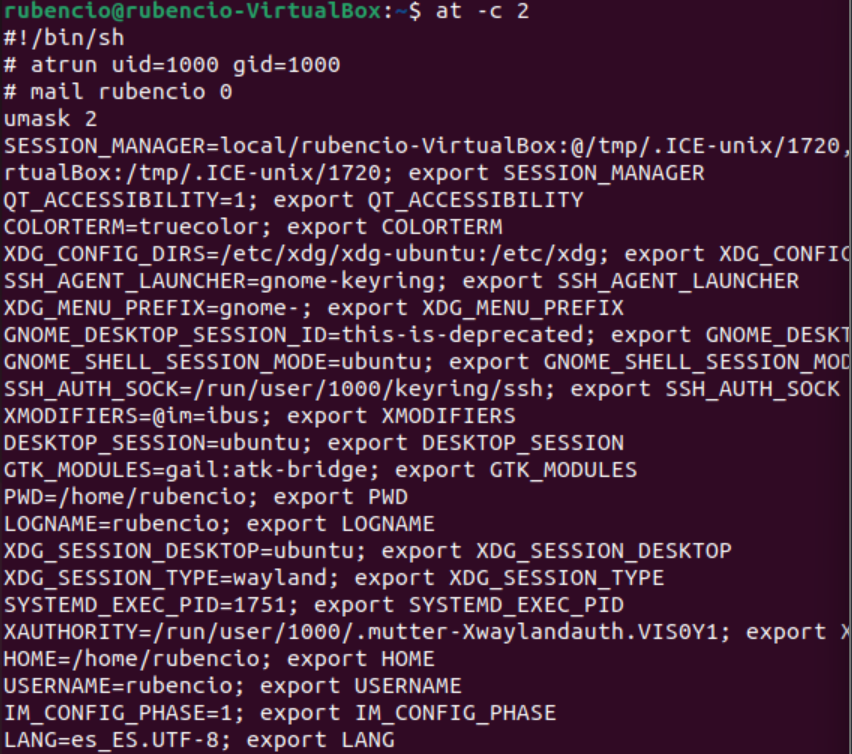


Mediante este comando se pueden automatizar tareas para realizar en un momento único especificando la hora, fecha y la tarea que deseamos realizar. La sintaxis de este comando es: at hora [fecha] [-f fichero].

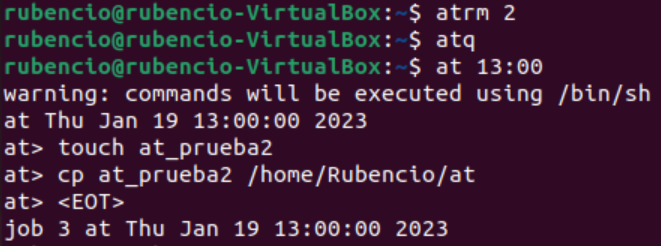


Las principales opciones de interés de este comando son las siguientes:

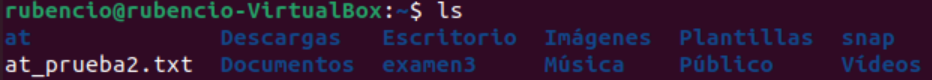
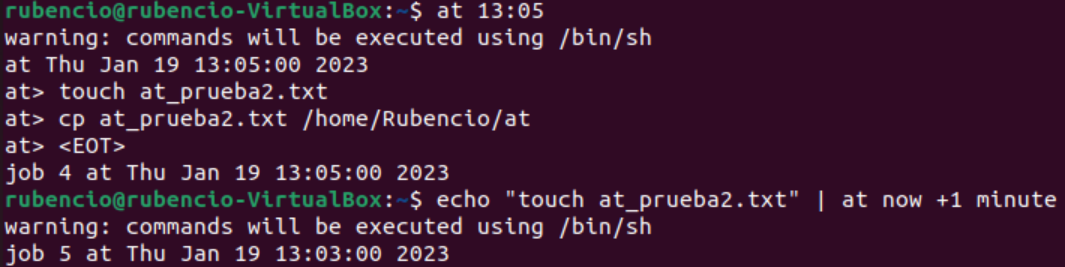
* Una de las características de este comando es la de recibir un correo electrónico al finalizar el proceso, esto se consigue mediante -m.
* Se pueden indicar tareas que se van a lanzar mediante un fichero con el modificador -f o por el contrario escribir la tarea y guardarla con el comando CTRL + R.
* Esta opción no es altamente destacable dado que muchos comandos cuentan con la misma función: at -l. De esta forma podemos listar los procesos listos para ejecutarse mediante at.
* De la mano de la opción anterior, tenemos los dos siguientes comando:
  + at -d num\_tarea: Para eliminar la ejecución de cualquiera de las tareas programadas.
  + at -c num\_tarea: Para ver el desglose de subtareas de una tarea programada

.

* Trasladando la visión a un punto de vista más amplio, se encuentra el comando para listar las tareas programas, pero a diferencia del comando at -l, este comando lista las tareas de todos los usuarios. Este comando es atq.
* 
* El siguiente comando sigue las características del comando anterior pero su función es similar a la de at -d num\_tarea. El comando atrm se encarga de eliminar las tareas programadas por cada usuario.

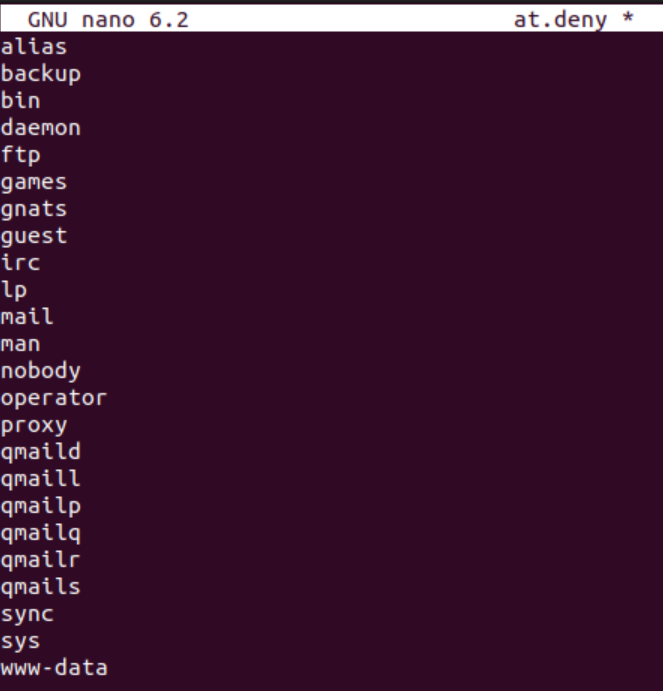


* Si lo que se busca es comprobar el contenido de la tarea programada se recurre al comando at -c, de esta manera se podrá comprobar los comandos que se encuentran en la tarea.
* Junto con el comando at se puede encontrar un “túnel de tiempo” que sirve para acelerar la tarea: echo “touch at\_prueba2.txt” | at now +1 minute.



* Una de las opciones más curiosas y destacadas de este comando es que se puede restringir el acceso al mismo. Esto no se realizará mediante comando sino que se tendrá que acceder al archivo que contiene todos los nombre de usuario que pueden trabajar con este. La ruta para acceder a este archivo es la siguiente: /etc/a.deny.

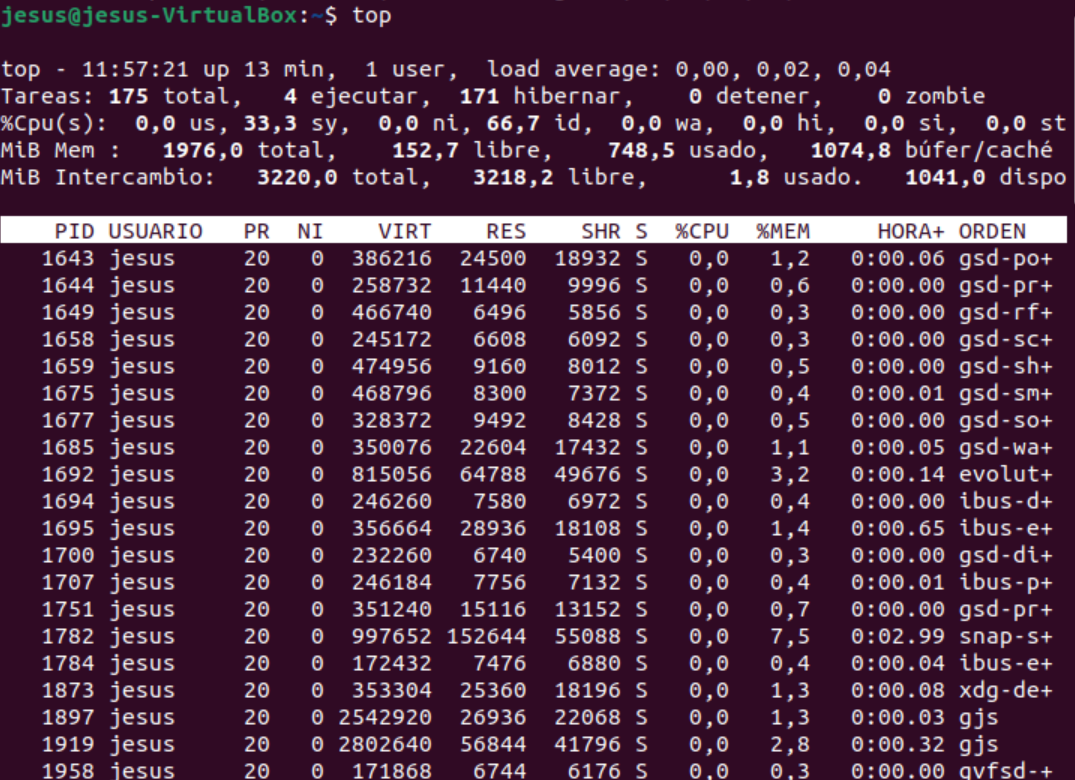


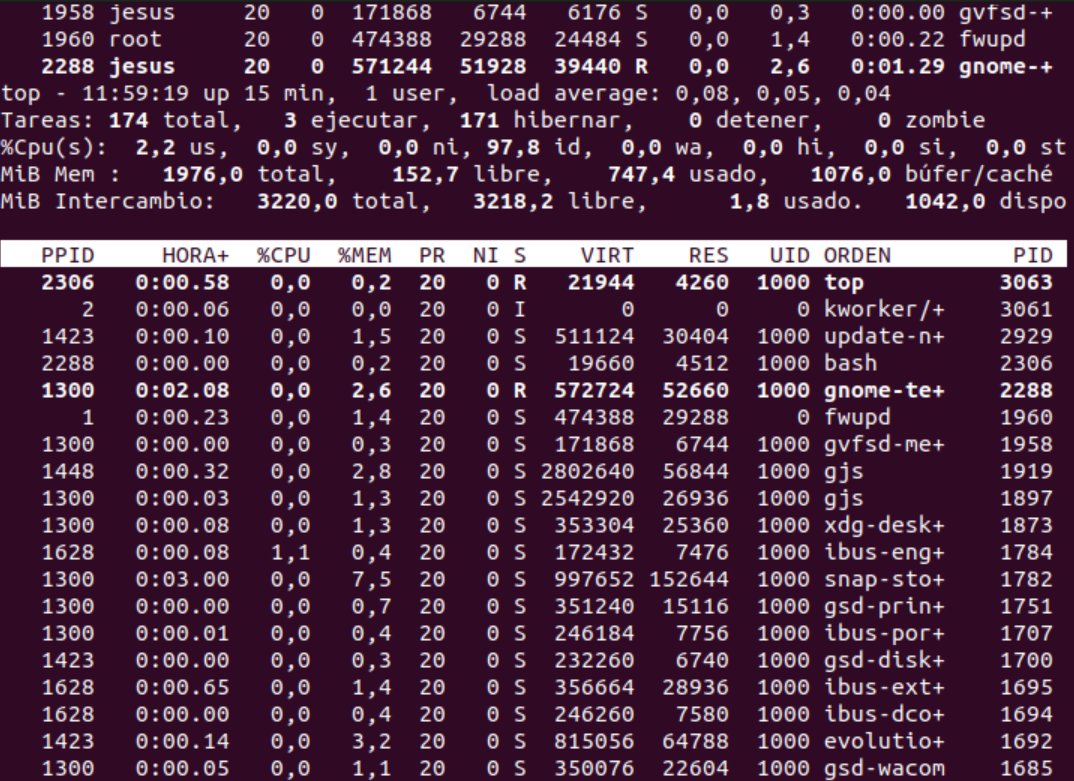


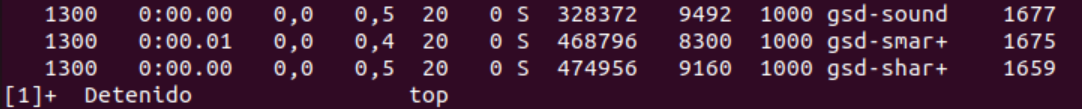
1. **Analizar y destacar las herramientas descritas en el apartado 4.7.**

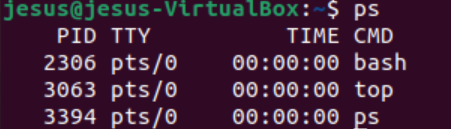


El comando *uptime* muestra la hora del sistema, el tiempo del sistema en activo, el número de usuarios y la carga del sistema en intervalos de 1,5 y 15 minutos. Equivale a la primera línea de la orden *top.*

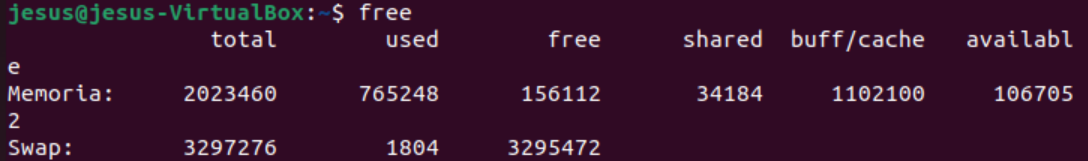
**

**

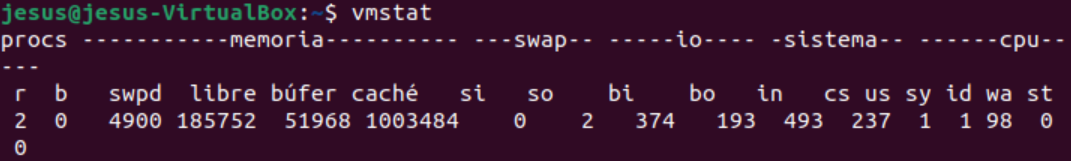


El comando ps se utiliza para mostrar por pantalla un listado de los procesos que se están ejecutando en el sistema

El comando *free* obtiene información sobre el espacio libre y usado de la memoria real y física.

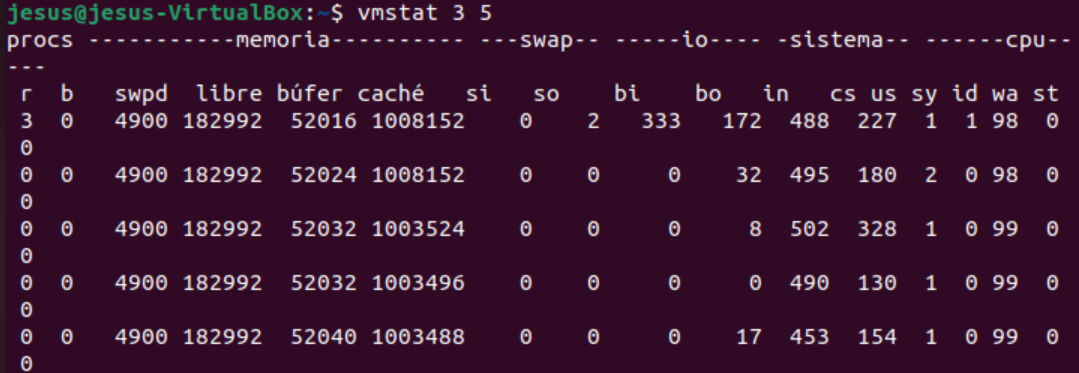


La herramienta *vmstat* devuelve la información de la memoria RAM, memoria virtual, intercambios entre memoria RAM y disco, interrupciones y el procesador.

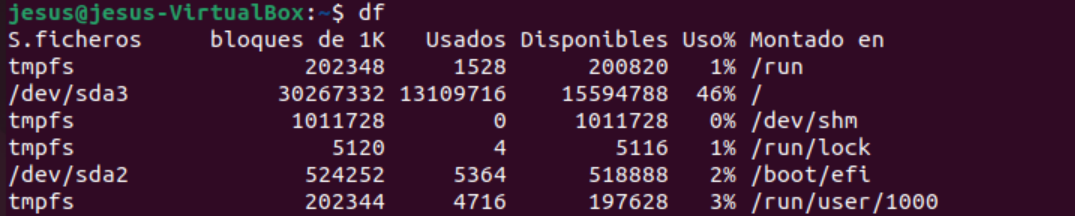


Su sintaxis es: *vmstat /tiempo /actualizaciones//,* donde:

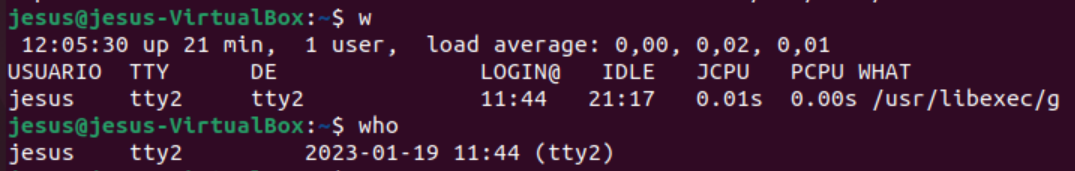
* tiempo: es el tiempo transcurrido entre dos actualizaciones
* actualizaciones: es el número de muestras. Como: *vmstat 3 5.*

**

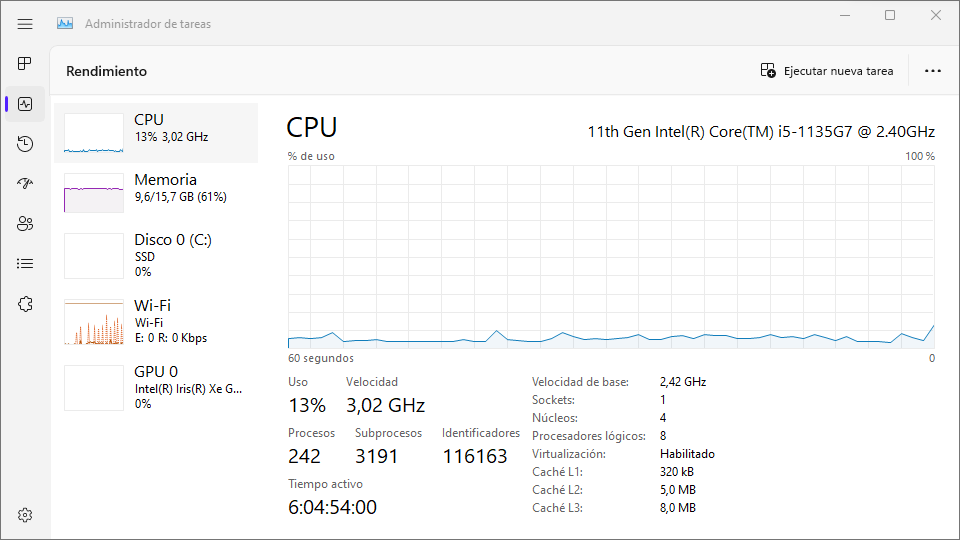
El uso del *df,* muestra el porcentaje de uso de la unidades de almacenamiento del sistema

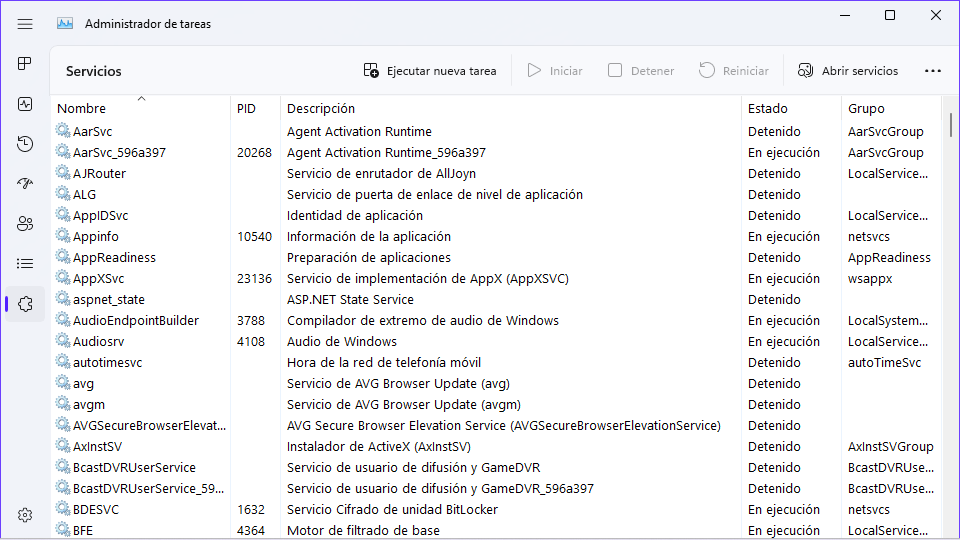
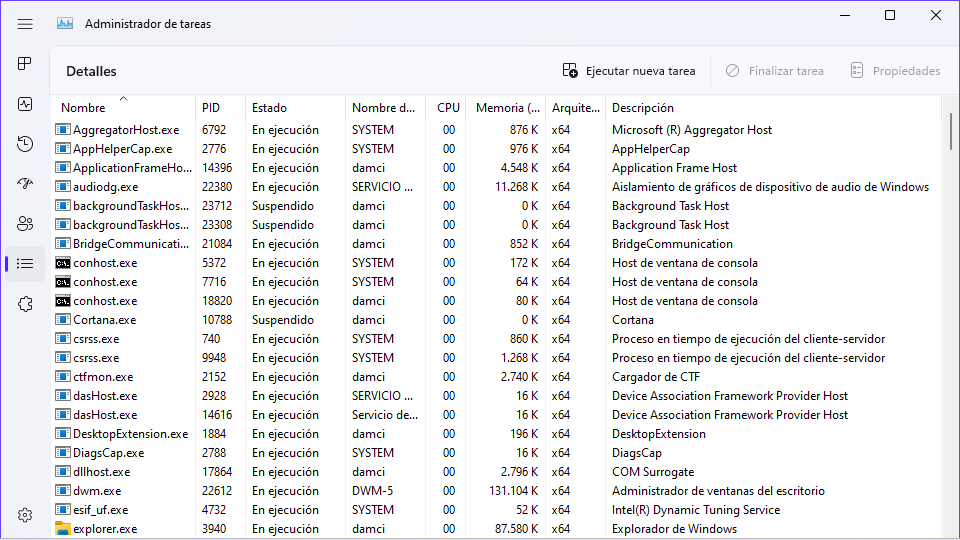


El comando w muestra quien esta conectado y que está haciendo (*parecido al* who)



En windows 11, a través del ‘Administrador de tareas’ podemos hacer un estudio preciso del ‘Rendimiento’ donde podemos analizar el uso de CPU, cantidad de procesos y subprocesos activos, estado de la memoria RAM, usos de los discos, tráfico de los adaptadores de red…



Además, podemos estudiar las aplicaciones en segundo plano y los servicios de windows en las pestañas ‘Detalles’ y ‘Servicios’

1. **Ampliar las aplicaciones descritas en el apartado 4.8.**

* **Aplicaciones de actualización y control de drivers.**
  + Drivereasy: Es de las más clásicas, tiene una interfaz sencilla en la cual te encuentras una sección de update(actualizar), otra de scan, otra de la información del hardware del equipo y otra de herramientas. Su uso principal es la actualización de driver aunque el inconveniente que tiene es que a no ser que pagues la suscripción la actualización de esto tienes que hacerla manualmente.
  + Driverbooster: Una aplicación muy parecida al drivereasy pero con la diferencia que la interfaz es más moderna y lo bueno de esta es que aunque tenga una versión de pago, con la versión gratis puedes hacer directamente la actualización de los drivers sin la necesidad de que sea manualmente.
  + Device doctor: Esta trae una interfaz muy sencilla. Lo malo de esta es que se limita a solo poder descargar un driver por día si no se instala la versión de pago, además con la versión de pago te trae también una opción para realizar un backup a los drivers por si instalaras uno que está mal pueda volver hacia atrás. Además con la versión de pago te trae un desinstalador, un limpiador de caché, un monitor del sistema para poder comprobar como está.
  + Slimdriver: Con una interfaz simple, en la cual las opciones son inició, explorar, resultados, opciones, restaurar, copias de seguridad y soporte. En principio no necesitas pagar para poder usar todas las opciones.
* **Aplicaciones de sincronización, copias de seguridad e imágenes del sistema.**
  + EaseUs Todo Backup Free:BackUp Maker almacena automáticamente sus archivos y al mismo tiempo ofrece una funcionalidad intuitiva.Viene con una interfaz de usuario sencilla pero con potentes habilidades para crear copias de seguridad del disco, la partición, el sistema operativo y los archivos y guardar las copias de seguridad en unidades locales, NAS, red y nube.
  + Clonezilla:Clonezilla es un programa de clonación de imágenes de partición y disco. Le ayuda a realizar la implementación del sistema, la copia de seguridad y la recuperación completa. Clonezilla guarda y restaura sólo los bloques usados en el disco duro, esto aumenta la eficiencia de la clonación.
  + FreeFileSync:Se trata de un programa gratuito y de código abierto que se utiliza para la sincronización de archivos. Es multiplataforma por lo que está disponible en Windows, Linux y OS X. Con FreeFyleSync podremos crear y gestionar copias de seguridad de nuestros archivos más importantes. Es un programa muy sencillo de usar ya que solo debemos de indicar las rutas de las dos carpetas que deseamos comparar.
  + Syncthing:Se trata de una aplicación que nos permitirá sincronizar archivos punto por punto entre dispositivos de una red local entre dispositivos remotos conectados a Internet.Syncthing está escrito en Go e implementa su propio protocolo de intercambio de bloques, igualmente libre, 2 ​ que permite generar una nube personal bajo el modelo BYO, donde los usuarios proporcionan el hardware y el software para la misma.
* **Optimización del sistema.**
  + Windows 10:
    - Desinstala aplicaciones que no utilices
    - Mantén limpio el escritorio del ordenador
    - Controla las aplicaciones que se ejecuten al inicio
    - Comprueba que el PC esté libre de malware
    - Libera espacio en tu disco duro
    - Desfragmenta el disco duro
    - Configuración memoria virtual
    - Cambia el plan de energía del ordenador
    - Menos efectos visuales
    - Quítale transparencias a la interfaz
    - Haz que Windows 10 se quede callado
    - Busca actualizaciones de sistema y controladores
    - Reinstala Windows 10 desde cero
    - No te olvides de limpiar tu torre
  + Linux:

- Hay unas cuantas maneras de optimizar un sistema Linux. Algunas cosas que puedes hacer son:

* + - Limpiar el sistema: Eliminar archivos y programas no deseados puede liberar espacio en disco y mejorar el rendimiento.
    - Actualizar el sistema: Asegurarse de tener las últimas actualizaciones de seguridad y software puede ayudar a mejorar el rendimiento.
    - Optimizar el arranque: Ajustar el orden de inicio de los programas y servicios puede mejorar el tiempo de inicio.
    - Ajustar la configuración del kernel: Ajustar la configuración del kernel para adaptarse a las necesidades del sistema puede mejorar el rendimiento.
    - Instalar un gestor de ventanas ligero: Utilizar un gestor de ventanas ligero en lugar de uno más pesado puede mejorar el rendimiento en sistemas con pocos recursos.
    - Utilizar un almacenamiento en caché: Utilizar un almacenamiento en caché puede mejorar el rendimiento al reducir el tiempo de acceso a disco.
    - Usar herramientas de optimización: Herramientas como "sysctl" o "tune2fs" pueden ayudar a optimizar el sistema.