****

**UNIVERSIDAD DE OVIEDO**

**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**INGENIERO DE TELECOMUNICACIÓN**

|  |
| --- |
| **LENGUAJES Y SISTEMAS INFORMÁTICOS** |

|  |  |
| --- | --- |
| **PROYECTO FIN DE CARRERA Nº** | **3133481** |

|  |
| --- |
| **GRABACION Y REPRODUCCION DE AUDIO-VIDEO EN UN PUESTO MULTIPANTALLA** |

|  |  |
| --- | --- |
| **DOCUMENTO Nº 5** |  |

|  |
| --- |
| **PRUEBAS Y RESULTADOS** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | **YÚSEF HABIB FERNÁNDEZ**  **OCTUBRE 2014**  **TUTOR: CLAUDIO DE LA RIVA**  **COTUTOR: MARCELINO AGUINAGA** | |
|  |  |  | |
|  |

TABLA DE CONTENIDOS

[1. INTRODUCCIÓN 5](#_Toc402943553)

[2. PRUEBAS 6](#_Toc402943554)

[Cronología de las pruebas 6](#_Toc402943555)

[Posible solución para la ejecución en cliente 7](#_Toc402943556)

[3. RESULTADOS 8](#_Toc402943557)

[3.1. Proceso de grabación 9](#_Toc402943558)

[3.2. Proceso de generación de archivos finales 11](#_Toc402943559)

[3.3. Reproducción de la sesión sobre el equipo original 12](#_Toc402943560)

TABLA DE IMAGENES

[Figura 1: Estado de la máquina en reposo 9](#_Toc402943163)

[Figura 2: Archivos genereados por cada uno de los flujos 10](#_Toc402943164)

[Figura 3: Entradas del archivo "txt" que se emplea para la concatenación 11](#_Toc402943165)

[Figura 4: Estado de la máquina tras lanzar las tres instancias de grabación 11](#_Toc402943166)

[Figura 5: Archivos finales almacenados en la carpeta de la sesión 12](#_Toc402943167)

[Figura 6: Estado de la máquina tras lanzar las dos instancias de generación de archivos finales 13](#_Toc402943168)

[Figura 7: Estado de la máquina tras lanzar la instancia de VLC para la reproducción de la sesión 14](#_Toc402943169)

**IDENTIFICACIÓN DEL DOCUMENTO**

|  |  |
| --- | --- |
| **Proyecto** | Grabación y reproducción de audio-video en un puesto multipantalla |
| **Nº proyecto** | 3133481 |
| **Autor** | Yúsef Habib Fernández |
| **Tutor** | Claudio de la Riva Álvarez |
| **Cotutor** | Marcelino Aguinaga Izquierdo |
| **Documento** | Pruebas y resultados |
| **Fecha** | Octubre 2014 |

# INTRODUCCIÓN

EL presente documento tiene la intención de mostrar las pruebas realizadas del prototipo desarrollado tanto a nivel local como en las instalaciones del cliente. Para ello se mostrará cronológicamente los diferentes pasos y pruebas realizadas. Finalmente se mostrará una posible solución para su implementación en el cliente.

A su vez este documento mostrará información relativa a los resultados obtenidos a nivel local del prototipo:

* Datos relevantes al tamaño de los archivos finales generados.
* Datos relevantes al sincronismo de los archivos temporales generados previos a la generación de los archivos finales.
* Datos relevantes al rendimiento del equipo durante la ejecución de las diferentes funcionalidades del prototipo desarrollado: consumo de CPU y uso de memoria durante las tareas más críticas.

# PRUEBAS

En el documento número 2 “Planificación y presupuesto”, se muestra la planificación y metodología del desarrollo de este prototipo software. Esta sección tiene la intención de centrarse en el proceso de realización de p

## Cronología de las pruebas

A continuación se detalla cronológicamente como se procedió para la realización de las pruebas del prototipo desarrollado. En la figura número X se muestra el equipo empleado para las pruebas locales.

IMAGENEQUIPOSOBREELQUESEDASARROLLA

* XXX: se comenzaron las pruebas sobre el equipo y sistema operativo (Linux Ubuntu) sobre el que se había desarrollado el prototipo. Estas primeras pruebas fueron positivas ya que se logró obtener las primeras grabaciones de los tres flujos audiovisuales en un equipo como se muestra en la imagen XXX
* XXX. Se exportó el código tal cual había funcionado anteriormente a los equipos del cliente. Esta primera prueba falló, debido a que estos no disponen de conexión a internet y los repositorios a los que tienen acceso son muy limitados por lo que no se logró descargar el software complementario necesario para el correcto funcionamiento del prototipo.
* XXX: Tras adaptar el código para poder llevar todos el software previamente descargado, se probó de nuevo el prototipo. Al igual que la vez anterior no se logró llegar a probar el software ya que no se cumplían todas las dependencias y no se lograba realizar una instalación satisfactoria.
* XXX: Se decidió por tanto, en buscar una solución que permitiese llevar configurados y compilados todos los paquetes necesarios. Para ello se empezó a emplear CentOs como sistema operativo de desarrollo, ya que es el S.O., más similar a Red Hat Enterprise, el cual es empleado por el cliente.
* XX: Se realizó la instalación de todo el software adicional y se comenzaron a generar archivos finales sobre un CentOs 5.0. Estos buenos resultados, conllevaron la decisión de configurar y compilar el software previamente antes de llevarlo al cliente.
* XX: La última prueba se realizó exportando todo este software preconfigurado y precompilado sobre CentOs, considerándose la única alternativa viable. Al igual que en los resultados obtenidos no fueron buenos, ya que aun siendo sistemas operativos muy similares, la estructura de los archivos de configuración y el árbol de directorios no permitieron el correcto funcionamiento.

## Posible solución para la ejecución en cliente

Las causas por las que no se logró hacer funcionar el prototipo correctamente en el cliente se muestran a continuación:

* Máquinas sin acceso a internet, por lo que no se puede actualizar el sistemas ni descargar las últimas versiones del software necesario de los repositorios oficiales de Red Hat Enterprise.
* El repositorios local al que tiene acceso la máquina, es el que viene incluido con el CD de instalación por lo que las versiones que ofrece tienen al menos 4 años de antigüedad que fue cuando se liberó la versión 5, que es la empleada.
* Red Hat Enterprise es un sistema operativo enfocado a las empresas por lo que es comercial. No facilitan binarios para su descarga por parte de un usuario, por lo que se buscó la versión libre más similar, CentOs. La cual tiene muchas cosas en común con Red Hat, pero por ejemplo al no compartir repositorios las versiones ofrecidas en uno y en el otro difieren.

Ante la necesidad de ofrecer una solución para la instalación del proyecto, se decidió optar por buscar una solución teórica. Para ello se consultó a uno de los desarrolladores del proyecto FFmpeg, el cual muy amablemente ofreció esta solución:

“*Desde una máquina con el mismo S.O. RHEL 5 y que tenga un disco duro parecido o el mismo que la máquina destino, realizar la configuración, compilación e instalación de todo el software necesario. A continuación y mediante CloneZilla realizar una copia de seguridad clon sobre un medio extraíble como un USB y restaurarla en la máquina destino. Es muy importante que ambos discos duros sean muy parecidos.”*

# RESULTADOS

Esta sección mostrará los resultados obtenidos por el prototipo durante las pruebas realizadas a nivel local. Para ello se harán análisis de los parámetros más determinantes como son los relacionados con los archivos generados, como por ejemplo su tamaño o sincronismo entre ellos. Y por otro lado se analizarán el gasto de recursos que genera la ejecución del prototipo en la máquina local.

El equipo empleado para las pruebas locales cuenta con cuatro núcleos por lo que el uso del 100% del CPU, se produce cuando se muestra un 400%. Por ejemplo en la imagen X, en la que se muestra el estado en reposo del equipo se observa como las tareas que se están ejecutando consumen un 2,9% del total del CPU y se tiene 2037852 KiB de memoria libre.

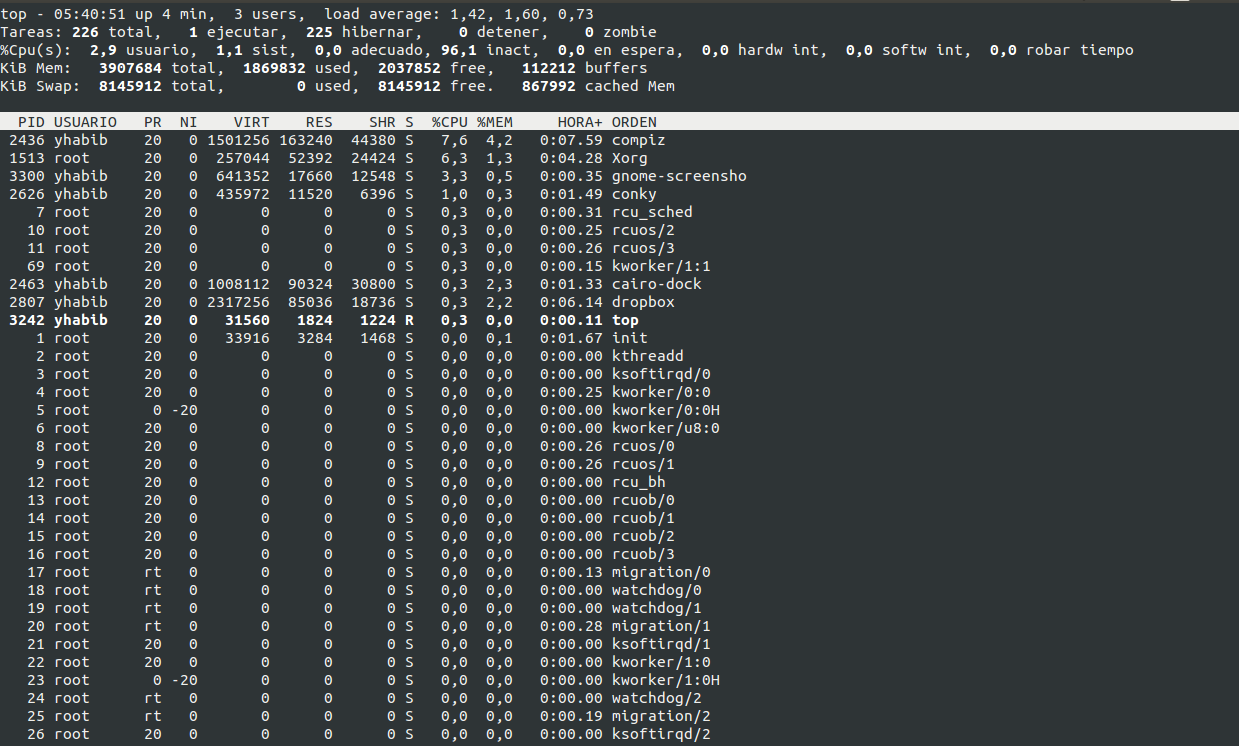


Figura 1: Estado de la máquina en reposo

## Proceso de grabación

El proceso de grabación comienza cuando las tres instancias de grabación de FFmpeg son lanzadas, y finaliza justo en el momento en que la sesión de grabación se ha completado. Este proceso y con la configuración generada por defecto “normal” genera tres archivos finales que son:

* Video.mp4, con un peso de X MB, contiene el video …
* Video.mp4 con un peso de Y MB y q contiene …
* Audio.mp3 con un peso de ZMb y que contiene….

Estos tres archivos se generan a partir de 10 archivos más pequeños cuya duración es especificada por el usuario en la configuración a emplear. En total 30 archivos que se almacenan en su carpeta correspondiente. En la figura 2 se puede se muestran los 10 archivos asociados a la grabación del flujo de audio, y el archivo “txt” contenedor de todas las entradas el cual será leído por el script “concat.sh” para poder generar el archivo final de audio, un ejemplo de este se puede encontrar en la figura 3.

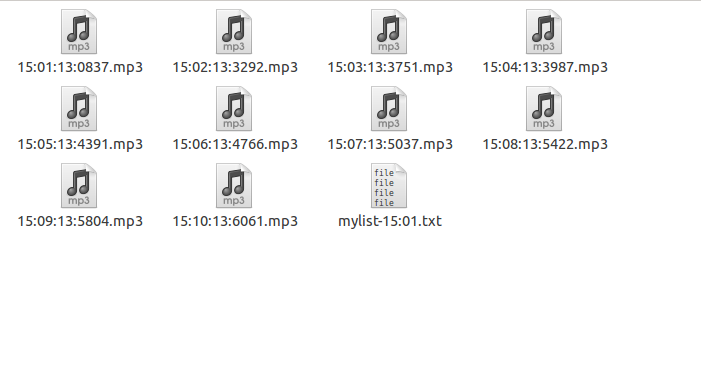


Figura 2: Archivos genereados por cada uno de los flujos

El retardo que se produce entre archivos, viene producido por el efecto de llamar a una nueva instancia de grabación cuando la actual acaba. El valor límite sería cero segundos, pero para lograr esto habría que emplear una configuración en la que solo se lanzase una instancia por lo que no se generarían múltiples archivos. Este proceso se descartó ya que incumplía las exigencias del cliente[[1]](#footnote-1).

La desincronización que se obtiene entre el primero de los archivos grabados y el último es de aproximadamente 500ms, para una duración total de 10 minutos por lo que representa un desfase casi nulo. En la siguiente imagen se puede observar este desfase a partir del archivo contenedor de todas las instancias de un flujo.

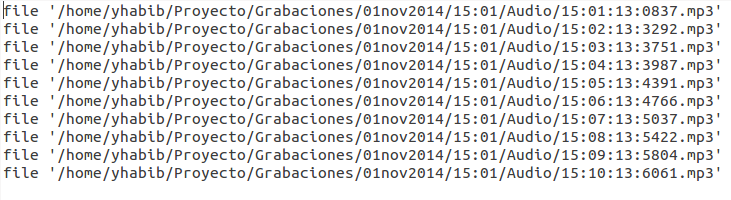


Figura 3: Entradas del archivo "txt" que se emplea para la concatenación

Respecto al gasto de recursos del sistema, se puede observar en la figura X, como ha aumentado como era de esperar, pero de una manera proporcionada. El incremento de gasto de CPU es de un 15% con tres instancias de grabación como son mostradas. La diferencia entre el gasto de cada una de ellas radica en que la del audio es la que menos procesado necesita, y de las de video, la encargada de manejar el monitor con mayor resolución será la que más gasto genere.

En cuanto al gasto de memoria, también se observa su crecimiento, pero de nuevo este es controlado, ya que no alcanza el 20%.

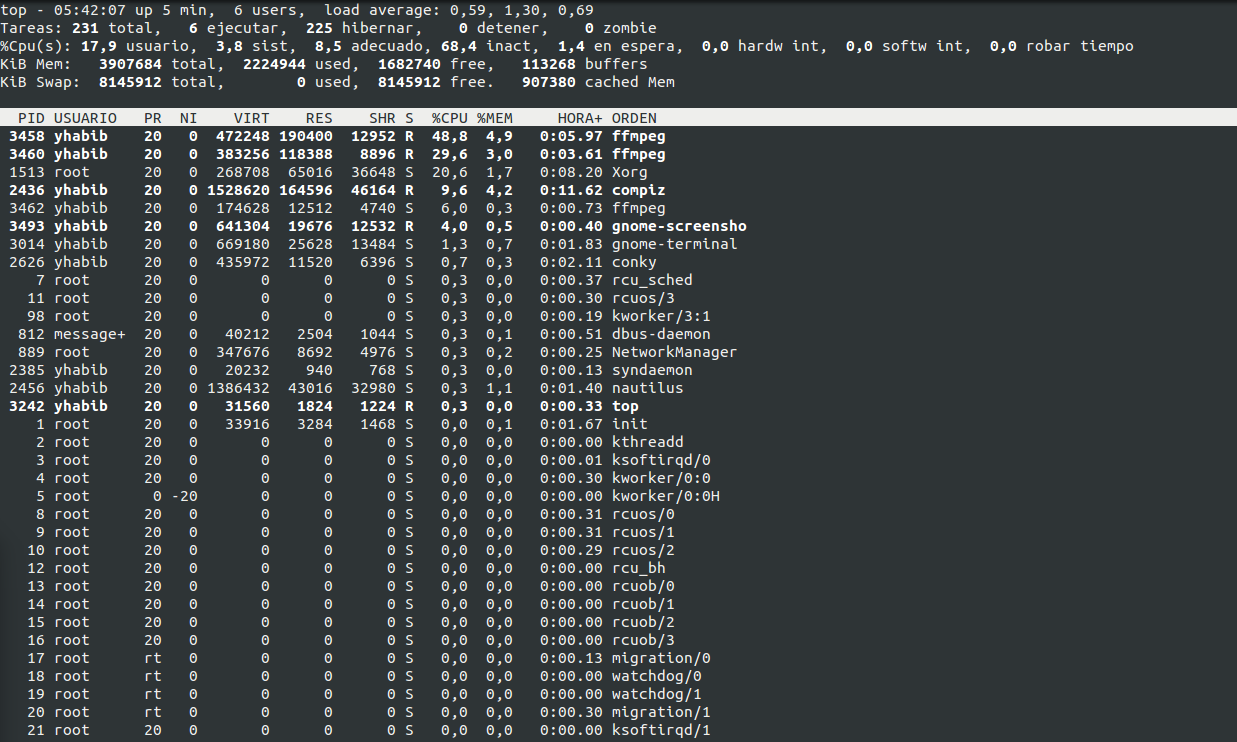


Figura 4: Estado de la máquina tras lanzar las tres instancias de grabación

## Proceso de generación de archivos finales

Se engloba dentro de este proceso, la concatenación del video del monitor principal con el audio, y la superposición de los tres flujos para la generación de un archivo final que incluya todo.

En la siguiente figura se muestran todos los archivos generados. Las carpetas contienen los fragmentos a partir de los cuales se crean los videos finales, estas presentan un aspecto como el mostrado en la figura 2.

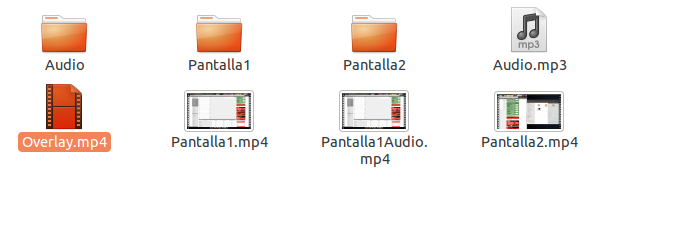


Figura 5: Archivos finales almacenados en la carpeta de la sesión

Es remarcable como el tamaño del archivo “Overlay.mp4” no llega a los 100Mb de peso para una grabación intensa como la que se realizó para este ejemplo. Esto quiere decir que el minuto de grabación de todo el flujo audiovisual, esto es los dos videos procedentes de cada uno de los monitores y el audio de entrada del micrófono requieren de menos de 10Mb por minuto de grabación.

Respecto al consumo de recursos del sistema ahora si se puede apreciar un incremento considerable pues las tareas de procesado de video consumen mucho. Como se observa en la figura 6 se ha producido un incremento del 44.4% respecto a la posición de reposo y del casi 30% respecto a la grabación. En cuanto al gasto de memoria esta también ha crecido siendo este incremento del 95%. Es importante destacar respecto a este valor que la configuración del prototipo permite que esta tarea sea realizada en un equipo externo a la propia posición REPRO.

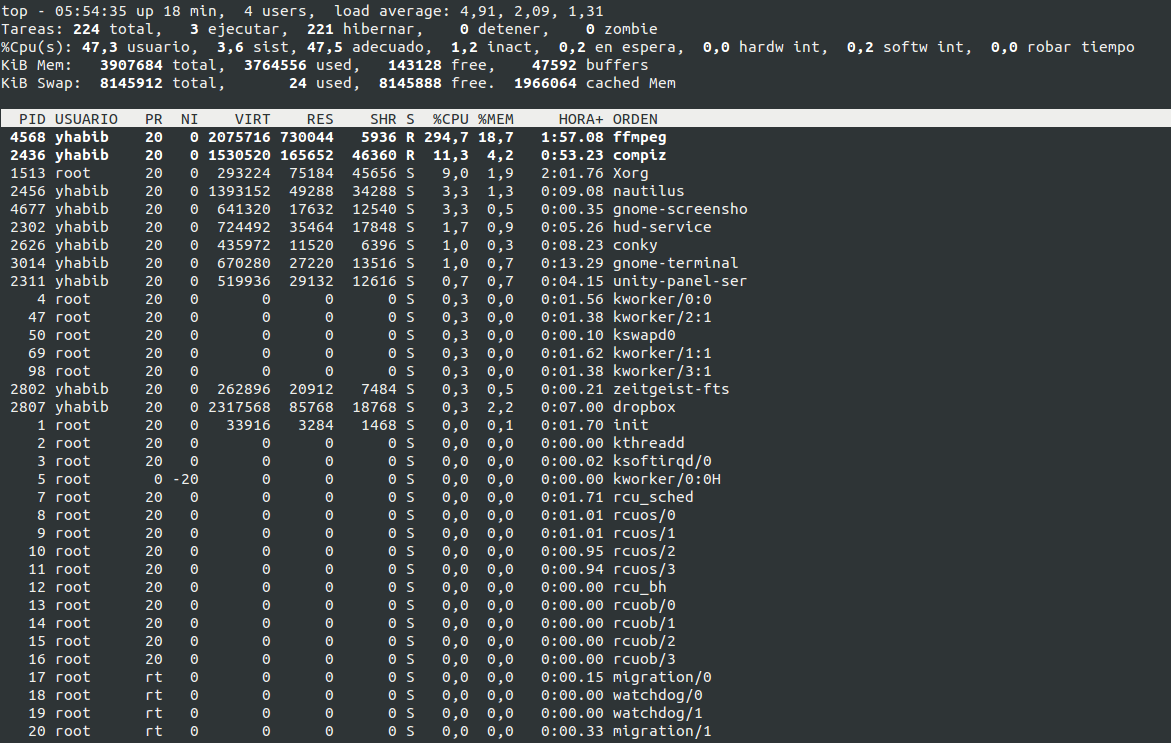


Figura 6: Estado de la máquina tras lanzar las dos instancias de generación de archivos finales

## Reproducción de la sesión sobre el equipo original

Finalmente se mostrará a continuación en la figura X el gasto de recursos del sistema asociados a la reproducción de una sesión sobre el equipo original. Esta viene determinada por el gasto de la instancia de VLC con el video del monitor primario y audio como entrada maestro, y el video del segundo monitor como entra esclava. Como se muestra en la imagen número 4, el gasto de CPU por parte del reproductor es bastante aceptable por cualquier máquina, mientras que el gasto de memoria al igual que pasó en el apartado previo se vuelve a disparar como consecuencia de la necesidad de procesar los 2 flujos audiovisuales a través de la instancia de VLC en modo “master-slave”.

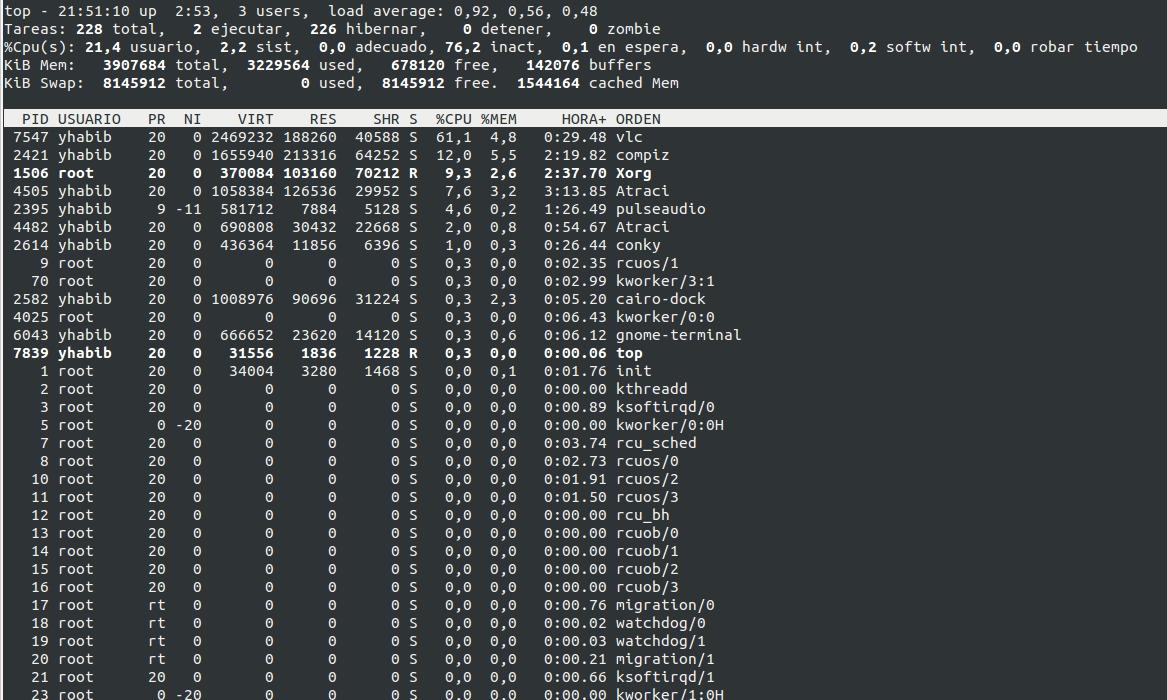


Figura 7: Estado de la máquina tras lanzar la instancia de VLC para la reproducción de la sesión

1. Se puede encontrar los requisitos del sistema en la sección 3 del documento número 1 “Memoria”. [↑](#footnote-ref-1)