

Screen splasher	Máxima puntuación: 5 puntos
Descripción:	
<p>Zeos actualmente tiene un único dispositivo pantalla que es compartido por todos los procesos. Este proyecto consiste en extender ZeOS para virtualizar este dispositivo y que cada proceso pueda usar un dispositivo pantalla en exclusiva para él.</p> <p>Además estos nuevos dispositivos deben interpretar algunos caracteres especiales que se le envíen para ejecutar diferentes operaciones (similar a https://wiki.bash-hackers.org/scripting/terminalcodes), permitiendo, como mínimo:</p> <ul style="list-style-type: none"> - cambiar la posición del cursor para situarse en cualquier posición de la pantalla - borrar el carácter actual - cambiar el color del carácter y el de fondo <p>Vamos a llamar <i>pantallas</i> a estos dispositivos. Cada proceso debe poder crear hasta 10 pantallas con una nueva llamada a sistema <i>createScreen()</i>, que devolverá un canal que puede ser usado por la llamada a sistema <i>write</i> existente. El valor devuelto será siempre el menor número disponible. Una pantalla puede destruirse con la llamada a sistema <i>close</i>. Y como máximo pueden crearse 30 pantallas</p> <p>El contenido de cada pantalla se guardará dentro del espacio de direcciones correspondientes a usuario, pero con protecciones de sistema (no accesibles por el usuario). Con esta nueva funcionalidad, cada proceso tendrá su propia pantalla (o pantallas) pero solo se podrá visualizar una en cada momento, por lo que hay que añadir el concepto de <i>foco</i>. Una pantalla sólo se visualizará cuando tenga el foco. Mediante la combinación de teclas 'TAB' y 'shift+TAB' debemos permitir el cambio de foco de una pantalla a la siguiente y a la anterior respectivamente. Para identificar cada pantalla debe mostrarse un identificador de pantalla y el identificador del proceso que la ha creado.</p> <p>También hay que añadir la llamada <i>int setFocus(int canal)</i> que debe permitir el cambio de foco inmediato a una pantalla específica del proceso que la ejecuta.</p> <p>Las consolas de un proceso deben heredarse al crear nuevos procesos.</p>	
1er nivel (máximo 2 puntos)	
Diseña e implementa el soporte mínimo necesario para poder tener múltiples pantallas dentro de un proceso. Para este primer nivel, solamente es necesario que el soporte funcione para un único proceso (creación pantallas y cambio de pantalla). Diseña e implementa juegos de prueba para comprobar el correcto funcionamiento.	
2º nivel (máximo 4 puntos)	
Diseña e implementa el soporte completo para que las consolas funcionen con múltiples procesos y se puedan crear y destruir. Diseña e implementa juegos de prueba para comprobar el correcto funcionamiento.	
3er nivel (máximo 5 puntos)	
<p>Para mejorar la eficiencia de ZeOS diseña e implementa un mecanismo de Copy-on-Write para retrasar la copia de las páginas de memoria. Su funcionamiento es el siguiente:</p> <ul style="list-style-type: none"> • Dentro de la llamada al sistema <i>fork</i>, en vez de copiar el contenido de las páginas físicas de un proceso a otro, se copia solamente la tabla de páginas de un proceso a otro. • Tanto en la tabla de páginas del proceso padre, como en la tabla de páginas del proceso hijo, las páginas físicas con permiso de escritura, se marcan como Copy-on-write y se les quita el permiso de escritura. • Cuando salta la excepción de page fault, se mira si el acceso es de escritura a una página marcada como copy-on-write y, en ese caso, se crea una copia de esa página para el proceso que ha provocado la excepción. Además, a la nueva página se le añade el permiso de escritura. 	