

◆ Bloque I: Sistema de ficheros

Nº	Comando	Descripción
I.1	<code>ls -lS doc</code>	Lista los archivos del directorio doc ordenados por tamaño (de mayor a menor).
I.2	<code>\ls -a</code>	<code>grep 'c\$'</code>
I.3	<code>find ??? -type f -name '*m'</code>	Busca archivos que terminan en m dentro de subdirectorios de 3 letras.
I.4	<code>find . -type d -regex '[ds].c' -exec ls {} \;</code>	Lista el contenido de subdirectorios de 3 letras que empiezan por d o s y terminan en c.
I.5	<code>mkdir ../test && cp scripts/*gnome* ../test && ls ../test</code>	Crea el directorio test, copia archivos con "gnome" desde scripts y lista su contenido.
I.6	<code>cp /usr/bin/cat ./mostrar</code>	Crea una copia del comando cat llamada mostrar en el directorio actual.
I.7	<code>./mostrar TODO && echo "*****" && cat TODO</code>	Ejecuta mostrar con TODO, imprime 10 asteriscos y luego muestra el contenido con cat.
I.8	<code>mv mostrar ../test && ls ../test</code>	Mueve mostrar al directorio test y muestra su contenido.
I.9	<code>cp -r src ../test/source-nano && ls -R ../test</code>	Copia recursivamente src a test/source-nano y lista todo su contenido.
I.10	<code>find sample_files/document/* -maxdepth 1 -type f \(-name '*5*.pdf' -o -name '*5*.rtf' \) -exec cp {} ../test \;</code>	Copia archivos que contienen "5" y terminan en .pdf o .rtf desde subdirectorios inmediatos de document a test.

◆ Bloque II: Búsquedas

Nº	Comando	Descripción
II.1	<code>find image document -type f -name '*e*e*' -size +600k -size -700k</code>	Busca archivos con dos "e" en el nombre y tamaño entre 600 y 700 KB.

Nº	Comando	Descripción
II.2	<code>find . -type d -regex '.*/*...[fg]'</code>	Encuentra directorios de 3 letras cuyo último carácter es f o g.
II.3	<code>grep -w 'id' document/txt/*.txt -l</code>	Busca archivos .txt que contienen la palabra id como palabra completa.
II.4	<code>grep -rn 'ergo.\$' .</code>	Busca líneas que terminan en “ergo” seguido de un carácter, mostrando archivo y línea.
II.5	<code>`grep -rl '^[li]' document/[123]</code>	<code>grep 't'</code>

◆ Bloque III: Procesos y redirecciones

Nº	Comando	Descripción
III.1	<code>`grep 'editor' NEWS</code>	<code>tail -n 2 > noticias`</code>
III.2	<code>`head -n 5 IMPROVEMENTS</code>	<code>grep -v '^\$'</code>
III.3	<code>find /etc -type f -name '*init' 2>/dev/null > ficheros_init</code>	Busca archivos que terminan en init en /etc, ignorando errores, y guarda en ficheros_init.
III.4	<code>sleep 1h & && jobs</code>	Lanza sleep en segundo plano por 1 hora y muestra los procesos activos.
III.5	<code>`pmap \$(pgrep sleep)</code>	<code>grep 'r-x'</code>
III.6	<code>pkill -9 sleep</code>	Mata todos los procesos sleep con señal SIGKILL.
III.7	<code>`grep -rl 'linux' src</code>	<code>xargs wc -l`</code>
III.8	<code>(grep -n 'brightblue' syntax/*java*)</code>	Muestra líneas con brightblue y su número en archivos relacionados con Java.

◆ Bloque IV: Git

Nº	Comando	Descripción
IV.1	<code>git restore AUTHORS && cat AUTHORS</code>	Recupera el archivo AUTHORS eliminado y muestra su contenido.
IV.2	<code>git diff README && git restore README</code>	Muestra diferencias del archivo README y descarta los cambios.

Nº	Comando	Descripción
IV.3	<code>git restore --staged doc/faq.html</code>	Saca el archivo <code>faq.html</code> del área de staging.
IV.4	<code>cd ~/recursos-examen/sample_files && git add LICENSE && git commit -m "Añadiendo LICENSE"</code>	Añade y confirma el archivo <code>LICENSE</code> en el repositorio.
IV.5	<code>git log --pretty=format:"Author: %an <%ae>" -- README*</code>	Muestra los autores que han modificado archivos que empiezan por <code>README</code> .
IV.6	<code>git rm README && git status && git commit -m "Eliminando README"</code>	Elimina <code>README</code> , muestra el estado y confirma la eliminación.

◆ Bloque V: Lenguaje ensamblador y filtros

Nº	Comando	Descripción
V.1	<code>gcc -static miprograma.c arrays_iguales.c -o compara_arrays_static && ls -l compara*</code>	Compila estáticamente y lista archivos que empiezan por <code>compara</code> .
V.2	<code>`ldd compara_arrays</code>	<code>grep libc</code>
V.3	<code>gcc -S arrays_iguales.c && grep 'cmp' arrays_iguales.s</code>	Compila a ensamblador y muestra instrucciones <code>cmp</code> .
V.4	<code>`objdump -d compara_arrays</code>	<code>grep call</code>
V.5	<code>`objdump -d compara_arrays_static</code>	<code>tail -n 1</code>
V.6	<code>`objdump -d /lib/x86_64-linux-gnu/libc.so.6</code>	<code>sed 's/IO_puts.cold/Rutina-libC-IOPuts-COLD/g'</code>