



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**



FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS

PRACTICA 2: DEPURACIÓN POR TRAZAS

PROFESOR: GUNNAR WOLF

NOMBRE: TIRADO PÉREZ JOSÉ JESÚS

NÚMERO DE CUENTA: 312135993

Para esta practica he utulizado el sistema operativo de Ubuntu de una distribucion unix ya que fue la opción de trabajo que nos presentó el profesor en clase y me pareció la mas adecuada. Sobre todo para entender el tema.

He utilizado la ejecución de un comando simple pero interesante. Se trata de **CP** que es para copiar un archivo en otro directorio dentro del equipo. Me gustó esta opción ya que es una de las operaciones comunmente usadas dentro de un sistema operativo para cualquier usuario y conocer a fondo como se lleva a cabo esta labor puede ser más que interesante.

Para realizar la traza del programa he utilizado la funcion de linux **strace**, los resultados los he copiado y trabajado sobre un editor de código llamado **SublimeText** ya que la herramienta me permite hacer visible la distincion entre los distos componentes que integran los resultados, como las propias funciones y sus argumentos, anexando tambien bloques de comentarios para cada una de las lineas que he analizado.

Dejo la captura de pantalla correspondiente a mi análisis.

La entrada en terminal fue: **jesus@jesus:~\$ strace cp /Desktop/prueba.txt /Downloads**

```

88 mprotect(0x7fe4b5b34000, 4096, PROT_READ) = 0
89 mprotect(0x7fe4b5d3c000, 4096, PROT_READ) = 0
90 mprotect(0x7fe4b5f5c000, 4096, PROT_READ) = 0
91 mprotect(0x623000, 4096, PROT_READ) = 0
92 mprotect(0x7fe4b6185000, 4096, PROT_READ) = 0
93 munmap(0x7fe4b616a000, 98881) = 0
94 set_tid_address(0x7fe4b6165ad0) = 25459
95 //es un atributo de un hilo generado por el proceso
96 set_robust_list(0x7fe4b6165ae0, 24) = 0
97 //solicita al kernel una lista robusta de hilos. El argumento escrito hace referencia a una solicitud de memoria (sizeof)
98 rt_sigaction(SIGRTMIN, {0x7fe4b4edbb50, [], SA_RESTORE|SA_SIGINFO, 0x7fe4b4ee7390}, NULL, 8) = 0
99 //es utilizada para cambiar la acción tomada, se manda una señal.
100 rt_sigaction(SIGRT_1, {0x7fe4b4edbbe0, [], SA_RESTORE|SA_RESTART|SA_SIGINFO, 0x7fe4b4ee7390}, NULL, 8) = 0
101 rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
102 //se utiliza para buscar o cambiar la mascara de la señal del hilo entrante. La mascara es el conjunto de señales cuya
103 entrega actualmente está bloqueada*/
104 getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
105 //obtiene límites de recursos, tiene dos límites. El límite suave es el valor
106 |que el núcleo hace para cumplir el recurso correspondiente
107 y el límite duro no aparece definido, aunque no sé a que haga referencia :( */
108 statfs("/sys/fs/selinux", {0x7ffe49e40680} = -1 ENOENT (No such file or directory)
109 /*devuelve informacion sobre el sistema de archivos montado, tiene como atributos el
110 Path de la dirección y un puntero a la estructura definida*/
111 statfs("/selinux", {0x7ffe49e40680} = -1 ENOENT (No such file or directory)
112 //forma parte del desglose de informacion sobre el sistema de archivos
113 brk(NULL) = 0x13b8000
114 //se cambia la ubicacion de la interrupcion.
115 brk(0x13d9000) = 0x13d9000
116 //se cambia la ubicacion de la interrupcion y por lo tanto tiene el efecto de asignar memoria al proceso.
117 open("/proc/filesystems", O_RDONLY) = 3
118 //abre al archivo con los permisos necesarios para la operación
119 fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
120 //devuelve la informacion del archivo al buffer apuntado
121 read(3, "nodev\tsysfs\nnodev\trootfs\nnodev\ttr"... , 1024) = 442
122 //lee el contenido asignado del descriptor de archivos
123 read(3, "", 1024) = 0
124 close(3) = 0
125 open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
126 //abre al archivo con los permisos necesarios para la operación
127 fstat(3, {st_mode=S_IFREG|0644, st_size=10219008, ...}) = 0
128 //devuelve la informacion del archivo al buffer apuntado
129 mmap(NULL, 10219008, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe4b4517000
130 close(3) = 0
131 geteuid() = 1000
132 //devuelve el id del usuario que realizo la llamada
133 stat("/Downloads", {0x7ffe49e40500} = -1 ENOENT (No such file or directory)
134 //muestra el estado del sistema de archivos
135 stat("/Desktop/prueba.txt", {0x7ffe49e40200} = -1 ENOENT (No such file or directory)
136 //muestra el estado del sistema de archivos
137 open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
138 //abre al archivo con los permisos necesarios para la operación
139 fstat(3, {st_mode=S_IFREG|0644, st_size=2995, ...}) = 0
140 //devuelve la informacion del archivo al buffer apuntado
141 read(3, "# Locale name alias data base.\n#"..., 4096) = 2995
142 //lee el contenido asignado del descriptor de archivos
143 read(3, "", 4096) = 0
144 //lee el contenido asignado del descriptor de archivos
145 close(3) = 0
146 open("/usr/share/locale/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
147 //abre al archivo con los permisos necesarios para la operación
148 open("/usr/share/locale/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
149 open("/usr/share/locale-langpack/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
150 open("/usr/share/locale-langpack/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = 3
151 fstat(3, {st_mode=S_IFREG|0644, st_size=619, ...}) = 0
152 mmap(NULL, 619, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe4b6182000
153 close(3) = 0
154 write(2, "cp: ", 4cp: ) = 4
155 write(2, "cannot stat '/Desktop/prueba.txt'...", 33cannot stat '/Desktop/prueba.txt') = 33
156 open("/usr/share/locale/en_US/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
157 open("/usr/share/locale/en_US/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
158 open("/usr/share/locale-langpack/en_US/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
159 open("/usr/share/locale-langpack/en_US/LC_MESSAGES/libc.mo", O_RDONLY) = -1 ENOENT (No such file or directory)
160 write(2, "No such file or directory", 27: No such file or directory) = 27
161 write(2, "\n", 1)
162 ) = 1
163 lseek(0, 0, SEEK_CUR) = -1 ESPIPE (Illegal seek)
164 close(0) = 0
165 close(1) = 0
166 close(2) = 0
167 exit_group(1) = ?
168 +++ exited with 1 +++
169

```