



Laboratorio 2 v2

Recuento de mensajes

ÍNDICE

1	OBJETIVOS.....	2
2	TOMA DE CONTACTO CON EL SERVIDOR DE INFORMACIÓN (IS).....	2
3	NÚMERO TOTAL DE MENSAJES ENVIADOS POR LOS PROCESOS.....	3
3.1	Definir y mantener el <i>array</i> de contadores de mensajes.....	3
3.2	Hacer que el núcleo pueda suministrar al IS el <i>array</i> de contadores.....	3
3.3	Conectar la nueva funcionalidad a la tecla de función F8.....	4
4	NÚMERO TOTAL DE MENSAJES ENVIADOS POR CADA PROCESO.....	5
5	MENSAJES ENVIADOS POR CADA PROCESO EN UN PERIODO DE TIEMPO.....	6

1 OBJETIVOS

Los principales objetivos de este Laboratorio son:

- Introducir en el *kernel* código para contar los mensajes enviados por los procesos
- Hacer visible a los procesos de usuario, a través del **IS** (Servidor de Información), los contadores de mensajes enviados por los procesos
- Mostrar por pantalla el número de mensajes enviados por los procesos, utilizando para ello la tecla de función **F9** que no está en uso

2 TOMA DE CONTACTO CON EL SERVIDOR DE INFORMACIÓN (IS)

MINIX 3 ofrece la posibilidad de obtener diferentes visualizaciones de información por pantalla a través de las teclas de función **F1-F12**. La tecla **F1** se usa para obtener un volcado con información acerca de la tabla de procesos y la tecla **F2** para obtener información sobre los mapas de memoria de los procesos. La tecla **F9** no tiene asignada ninguna funcionalidad.

Durante la inicialización, el **IS** registra la asociación de las teclas de función con las funciones encargadas de su tratamiento mediante una tabla denominada **hooks** tal y como veremos más adelante.

En este primer apartado lo que haremos será asociar la tecla de función **F9** con una funcionalidad ya existente “*kernel parameters*” que está asociada a la tecla de función **F10**. De esta forma, si pulsamos **F9** debe darnos la misma información que si hubiésemos pulsado la tecla **F10**.

Las teclas de función están definidas en **/usr/src/servers/is/dmp.c**. Si echamos un vistazo a este fichero y buscamos la tabla **hooks**, veremos que efectivamente la tecla **F9** no tiene, actualmente, asignada ninguna funcionalidad y lo fácil que parece incluirla en dicha tabla para que haga lo mismo que hace **F10**. Observar que el número de enganches definidos (**NHOOKS**) se realiza automáticamente, por lo tanto no hace falta modificarlo para que se tenga en cuenta que hemos incluido una nueva entrada en la tabla **hooks**.

- En este Laboratorio hay que partir de la imagen original de MINIX.
- Entrar en MINIX.
- Crear un directorio para el desarrollo del Laboratorio tecleando: **mkdir lab2**. En este directorio podemos copiar los ficheros originales y los modificados para poder volver a la versión original y tener una copia de lo que vayamos modificando.
- Editar el fichero **/usr/src/servers/is/dmp.c** y asignar a la tecla **F9** la misma funcionalidad que la tecla **F10** siguiendo las indicaciones dadas anteriormente.
- Cambiarse al directorio **/usr/src/releasetools** y teclear: **make install**.
- Arrancar con la nueva imagen: **halt + menu** (opción 3).

- Comprobar que la tecla de función **F9** tiene asociada la misma funcionalidad que la tecla de función **F10**.

3 NÚMERO TOTAL DE MENSAJES ENVIADOS POR LOS PROCESOS

El objetivo de este apartado es contabilizar los mensajes que envía cada proceso (incluidas las tareas) y, mediante la tecla de función **F9**, informar del número total de mensajes enviados entre todos ellos, para así tener una primera idea de la actividad global del sistema.

El desarrollo completo de este apartado, que supone tocar nueve ficheros, lo vamos a escalonar en tres partes:

1. Definir y mantener el *array* de contadores de mensajes
2. Hacer que el núcleo pueda suministrar al **IS** este *array* de contadores
3. Conectar la tecla de función **F9** con una funcionalidad que sume el *array* de contadores y muestre por pantalla el total de mensajes enviados por los procesos

3.1 Definir y mantener el *array* de contadores de mensajes

Registrar la antigüedad de un proceso, para llevar la cuenta de cuántos mensajes envía cada proceso declararemos un *array* del mismo tamaño que la tabla de procesos y de tipo “**long**” ya que, como veremos más adelante, pueden llegar a registrarse muchos envíos de mensajes. A este *array* le denominaremos **numMensajes** y se definirá en el fichero **/usr/src/kernel/proc.h**.

El contador de mensajes enviados por un proceso debe incrementarse cada vez que se llame al procedimiento **mini_send** que se encuentra en el fichero **/usr/src/kernel/proc.c**. Si quisiéramos hacer un control más exacto de los mensajes enviados, deberíamos tener en cuenta cuáles son realmente recibidos pero, por simplicidad, vamos a mantener la solución indicada.

- Editar el fichero **/usr/src/kernel/proc.h** y definir el *array* **numMensajes**.
- Editar el fichero **/usr/src/kernel/main.c** y dar el valor inicial 0 a todos los elementos del *array* **numMensajes**.
- Editar el fichero **/usr/src/kernel/proc.c** y tocar el procedimiento **mini_send** para incrementar el contador de mensajes correspondiente.
- Cambiarse al directorio **/usr/src/releasetools** y teclear: **make install**.
- Arrancar con la nueva imagen: **halt + menu** (opción 3).
- Comprobar que el sistema, aparentemente, sigue funcionando correctamente.

3.2 Hacer que el núcleo pueda suministrar al IS el array de contadores

Para que el *array* de contadores, residente en el espacio del núcleo, pueda ser reclamado por el IS, el núcleo ofrece una llamada al sistema “**sys_getinfo**” que ya proporciona mucha información del sistema y que podemos ampliar para que ahora también permita recuperar el *array* de contadores de mensajes.

Los servicios de información disponibles a través de “**sys_getinfo**” pueden consultarse en el fichero `/usr/src/include/minix/com.h` bajo el epígrafe “*Field names for SYS_GETINFO*”, que define constantes como `GET_PROCTAB` para indicar que se desea reclamar la información de la tabla de procesos del núcleo. Nosotros definiremos una nueva constante que llamaremos `GET_NMENSAJES`.

La función “**sys_getinfo**” se implementa en el fichero `/usr/src/kernel/system/do_getinfo.c` donde se contempla –dentro de una sentencia *switch*– cada uno de los distintos tipos de información que se le pueden solicitar. Nosotros deberemos incluir, como un nuevo caso, el tratamiento para devolver el *array* de contadores de mensajes. El esqueleto de este nuevo trozo de código sería:

```
case GET_NMENSAJES: {
    length    = <tamaño-del-array-de-contadores>;
    src_vir   = vir2phys(<dirección-del-array>);
    break;
}
```

En `<tamaño-del-array-de-contadores>` se tiene que indicar el tamaño –en bytes– de nuestro *array* de contadores, y como parámetro de `vir2phys` la dirección física de dicho *array* “**numMensajes**” para que luego se envíe al solicitante con `phys_copy`.

Finalmente, resaltar que en el fichero `/usr/src/include/minix/syslib.h` se declaran sinónimos cortos para invocar la función “**sys_getinfo**” particularizada para cada uno de los tipos de información que se desean reclamar. En nuestro caso deberemos incluir una línea como la siguiente: `#define sys_getnmensajes(dst) sys_getinfo(GET_NMENSAJES, dst, 0,0,0)`

- Editar el fichero `/usr/src/include/minix/com.h` y definir la nueva constante `GET_NMENSAJES`. (Con valor 25, Luego de la 24 : `GET_REGS`)
- Editar el fichero `/usr/src/kernel/system/do_getinfo.c` e incluir el tratamiento del nuevo caso: `case GET_NMENSAJES: {`
- Editar el fichero `/usr/src/include/minix/syslib.h` e incluir el sinónimo corto `sys_getnmensajes(dst)` apuntando a nuestro “`GET_NMENSAJES`” con parámetros `dst, 0,0,0`.
- Cambiarse al directorio `/usr/src/releasetools` y teclear: `make install`.
- Arrancar con la nueva imagen: `halt + menu` (opción 3).
- Comprobar que el sistema, aparentemente, sigue funcionando correctamente.

3.3 Conectar la nueva funcionalidad a la tecla de función F9

Para poder probar que el código introducido hasta ahora es correcto, vamos a conectar la tecla de función **F9** con un procedimiento que muestre en pantalla el número total de mensajes enviados por los procesos. Básicamente, este procedimiento debería solicitar al *kernel* el *array* de contadores a través de la función **sys_getnmensajes** que ya tenemos implementada, sumar el contenido de dicho *array* y mostrar el total por pantalla.

Los procedimientos invocados por las teclas de función están definidos en el fichero **/usr/src/servers/is/dmp_kernel.c** y siguen la sintaxis **xxxx_dmp**, como por ejemplo **proctab_dmp**. Por lo tanto, denominaremos **nmensajes_dmp** al procedimiento que conectaremos con **F9**.

Para poder definir el procedimiento **nmensajes_dmp** en un fichero y asociarlo a una tecla en otro, es necesario definir un prototipo para el mismo en **/usr/src/servers/is/proto.h** junto con el resto de prototipos ya declarados para **dmp_kernel.c**.

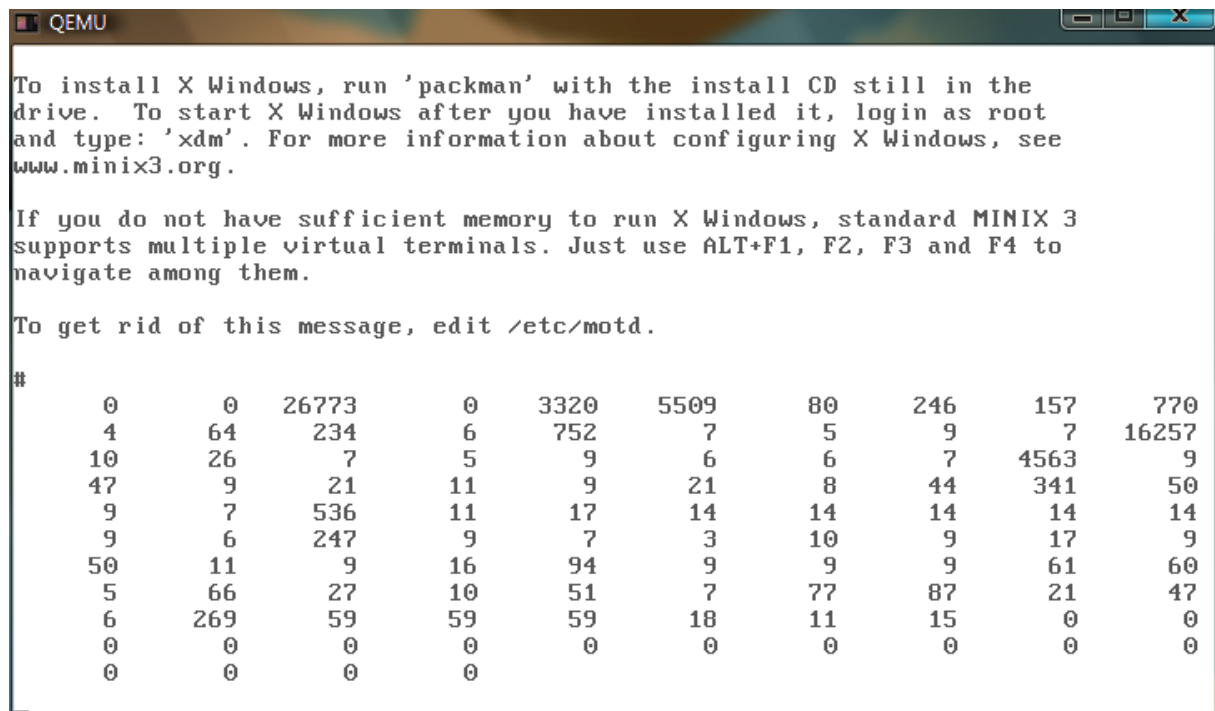
- Editar el fichero **/usr/src/servers/is/proto.h** y definir un nuevo prototipo para la función **nmensajes_dmp(void)**. (En en la sección */*dmp_kernel.c */*)
- Editar el fichero **/usr/src/servers/is/dmp_kernel.c** e implementar el procedimiento **nmensajes_dmp** que ha de mostrar por pantalla el número total de mensajes enviados por todos los procesos.
- Editar el fichero **/usr/src/servers/is/dmp.c** y enganchar la tecla **F9** al nuevo procedimiento **nmensajes_dmp**.
- Cambiarse al directorio **/usr/src/releasetools** y teclear: **make install**.
- Arrancar con la nueva imagen: **halt + menu** (opción 3).
- Pulsar la tecla **F9**. Debe aparecer un número de mensajes bastante elevado, puede que varias decenas de miles –depende mucho del tiempo que tarde en pulsarse **F9**–.
- Ahora vamos a estimar la cantidad de mensajes que se están enviando aunque el sistema esté aparentemente en inactividad. Para ello, vamos a pulsar cinco veces seguidas la tecla **F9** equiespaciadas 30 segundos. Rellenar la tabla siguiente:

Instante	Número de mensajes	Diferencia
T_0		$T_i - T_{i-1}$
T_0+30''		
T_0+60''		
T_0+90''		
T_0+120''		

- Calcular, aproximadamente, cuántos mensajes se envían por segundo:

4 NÚMERO TOTAL DE MENSAJES ENVIADOS POR CADA PROCESO

Ahora vamos a intentar ver qué procesos son los responsables de esta elevada tasa de envío de mensajes incluso en una fase de aparente inactividad. Para ello, vamos a modificar el procedimiento **nmensajes_dmp** para que muestre, realmente, todo el contenido del array de contadores **numMensajes** y no sólo el total. La salida esperada es como la que se muestra a continuación:



```

To install X Windows, run 'packman' with the install CD still in the
drive. To start X Windows after you have installed it, login as root
and type: 'xdm'. For more information about configuring X Windows, see
www.minix3.org.

If you do not have sufficient memory to run X Windows, standard MINIX 3
supports multiple virtual terminals. Just use ALT+F1, F2, F3 and F4 to
navigate among them.

To get rid of this message, edit /etc/motd.

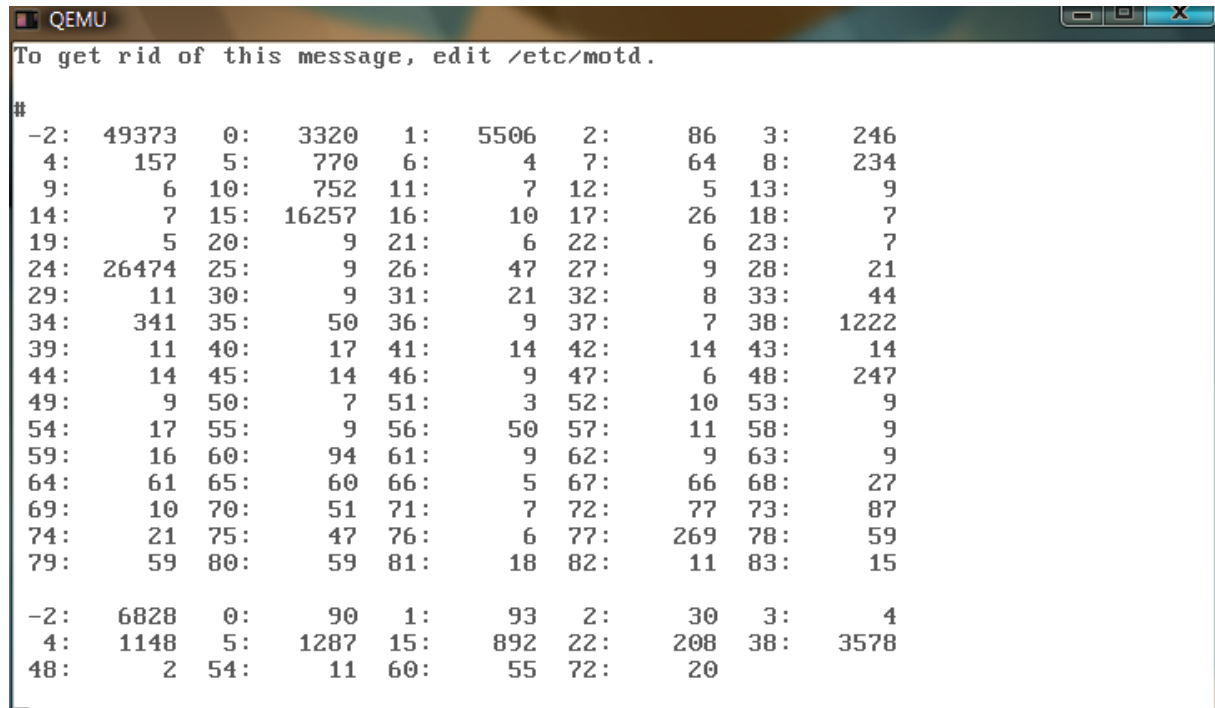
#
  0      0  26773      0  3320  5509      80   246   157   770
  4     64   234      6   752    7      5     9     7 16257
 10     26     7      5     9     6     6     7  4563     9
 47     9     21     11     9    21     8    44   341   50
  9     7   536     11    17    14    14    14    14    14
  9     6   247      9     7     3    10     9    17     9
 50    11     9    16    94     9     9     9    61    60
  5    66    27    10    51     7    77    87    21    47
  6   269    59    59    59    18    11    15     0     0
  0     0     0     0     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0

```

- Guardar la versión actual del fichero **/usr/src/servers/is/dmp_kernel.c** en el directorio lab2 con el nombre **dmp_kernelA.c**.
- Editar el fichero **/usr/src/servers/is/dmp_kernel.c** y modificar el procedimiento **nmensajes_dmp** para que ahora muestre una pantalla como la de la figura anterior.
- Cambiarse al directorio **/usr/src/releasetools** y teclear: **make install**.
- Arrancar con la nueva imagen: **halt + menu** (opción 3).
- Pulsar la tecla **F9** y comprobar si el resultado es correcto.
- Pulsar dos veces seguidas la tecla **F9** espaciadas 30 segundos y anotar los dos procesos que generan el mayor número de mensajes en ese intervalo de tiempo:
- La explicación del número de mensajes, en parte, tiene que ver con la interrupción periódica que se genera 60 veces por segundo.

5 MENSAJES ENVIADOS POR CADA PROCESO EN UN PERIODO DE TIEMPO

Para poder analizar con más facilidad el número de mensajes enviados como respuesta a algún tipo de comando, lo que haremos es modificar el procedimiento **nmensajes_dmp** para que muestre los mensajes enviados por los procesos entre pulsaciones consecutivas de la tecla **F9**. Aquellos procesos que no hayan enviado mensajes no se mostrarán, razón por la que debe indicarse el identificador de cada proceso mostrado. La salida esperada, tras pulsar dos veces **F9**, debe ser como la que se muestra a continuación:



```

QEMU
To get rid of this message, edit /etc/motd.

#
-2: 49373 0: 3320 1: 5506 2: 86 3: 246
4: 157 5: 770 6: 4 7: 64 8: 234
9: 6 10: 752 11: 7 12: 5 13: 9
14: 7 15: 16257 16: 10 17: 26 18: 7
19: 5 20: 9 21: 6 22: 6 23: 7
24: 26474 25: 9 26: 47 27: 9 28: 21
29: 11 30: 9 31: 21 32: 8 33: 44
34: 341 35: 50 36: 9 37: 7 38: 1222
39: 11 40: 17 41: 14 42: 14 43: 14
44: 14 45: 14 46: 9 47: 6 48: 247
49: 9 50: 7 51: 3 52: 10 53: 9
54: 17 55: 9 56: 50 57: 11 58: 9
59: 16 60: 94 61: 9 62: 9 63: 9
64: 61 65: 60 66: 5 67: 66 68: 27
69: 10 70: 51 71: 7 72: 77 73: 87
74: 21 75: 47 76: 6 77: 269 78: 59
79: 59 80: 59 81: 18 82: 11 83: 15

-2: 6828 0: 90 1: 93 2: 30 3: 4
4: 1148 5: 1287 15: 892 22: 208 38: 3578
48: 2 54: 11 60: 55 72: 20
  
```

Para poder mostrar sólo el número de mensajes enviados entre dos activaciones seguidas de la tecla **F9**, se recomienda guardar en otro *array*, que podemos denominar **numMensajesAnt**, el número total de mensajes enviados hasta ahora. Recordar que este *array*, declarado como global, puede inicializarse en tiempo de compilación como:

```

PUBLIC long numMensajesAnt [NR_TASKS+NR_PROCS] = {
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    -----,
    0, 0, 0, 0
}
  
```

- Guardar la versión actual del fichero **/usr/src/servers/is/dmp_kernel.c** en el directorio practica4 con el nombre **dmp_kernelB.c**.
- Editar el fichero **/usr/src/servers/is/dmp_kernel.c** y modificar el procedimiento **nmensajes_dmp** para que ahora muestre una pantalla como la de la figura anterior.
- Cambiarse al directorio **/usr/src/releasetools** y teclear: **make install**.
- Arrancar con la nueva imagen: **halt + menu** (opción 3).

- Pulsar la tecla **F9** y comprobar si el resultado es correcto.
- Después de unos 30 segundos volver a pulsar **F9**.
- Pulsar la tecla **<Enter>** y a continuación **F9**. Identificar qué nuevos procesos aparecen ahora que no aparecían antes y con cuántos mensajes.
- Volver a pulsar la tecla **<Enter>** ahora dos veces seguidas y a continuación **F9**. Explicar qué pasa –desde el punto de vista de envío de mensajes- cuando se pulsa la tecla **<Enter>**.
- Repetir la misma prueba para otro tipo de comandos tales como: *echo*, *sleep*, etc.
- Guardar la versión actual del fichero `/usr/src/servers/is/dmp_kernel.c` en el directorio lab2 con el nombre **dmp_kernelC.c**

Metodología

- El trabajo se realiza en grupos de **3** integrantes como máximos.
- Se debe entregar un informe donde se describan los procedimientos realizados, capturas de pantalla de los resultados, o información que se considere relevante.
- Los informes serán entregados vía email en formato PDF. Se debe incluir el nombre de los integrantes del grupo.
- Obligatorio realizar hasta el **punto 3**. Puntos 4 y 5 son opcionales.
- Fecha límite de entrega: Recuperatorio primer parcial.