

Nivel 13: verificacion.c

En este nivel se creará un nuevo programa, [verificacion.c](#), que recorrerá secuencialmente el fichero "prueba.dat" de cada proceso y generará la siguiente información:

- **Proceso:** PID que aparece en el nombre del directorio.
- **Número de escrituras:** contador de los registros *validados* dentro del fichero "prueba.dat" (se validan verificando que el campo PID coincida con el del proceso, ya que podríamos haber leído basura).
- **Primera escritura:** registro con el nº de escritura menor.
- **Última escritura:** registro con el nº de escritura mayor.
- **Menor posición:** registro con la posición (nº registro) más baja.
- **Mayor posición:** registro con la posición (nº registro) más alta.

Es decir de las 50 escrituras solo mostraremos por pantalla las 4 más significativas: primera y última escritura, menor y mayor posición.

```
//verificacion.h
#include "simulacion.h"

struct INFORMACION {
    int pid;
    unsigned int nEscrituras; //validadas 1
    struct REGISTRO PrimeraEscritura;
    struct REGISTRO UltimaEscritura;
    struct REGISTRO MenorPosicion;
    struct REGISTRO MayorPosicion;
};
```

Para cada una de esas escrituras se mostrará: el número de escritura, el nº de registro y la fecha. Ejemplo:

| | | | |
|--------------------------|----|--------|--------------------------|
| PID: 8145 | | | |
| Numero de escrituras: 50 | | | |
| Primera Escritura | 1 | 452816 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 246650 | Thu May 26 18:34:22 2022 |
| Menor Posición | 19 | 8981 | Thu May 26 18:34:21 2022 |
| Mayor Posición | 13 | 492815 | Thu May 26 18:34:20 2022 |

¹ El número de escrituras, por regla general, tiene que valer **50** (a menos que haya casualmente algún solapamiento).

Algoritmo detallado:

```
Comprobar la sintaxis. //Uso: verificacion <nombre_dispositivo> <directorio_simulaci3n>
Montar el dispositivo virtual.
Calcular el n3 de entradas del directorio de simulaci3n a partir del stat de su inodo. //100
Si numentradas != NUMPROCESOS entonces ERROR fsi.
Crear el fichero "informe.txt" dentro del directorio de simulaci3n.
Leer los directorios correspondientes a los procesos. //Entradas del directorio de simulaci3n
Para cada entrada de directorio de un proceso hacer
    Leer la entrada de directorio. 2
    Extraer el PID a partir del nombre de la entrada y guardarlo en el registro info. 3
    //Recorrer secuencialmente el fichero prueba.dat utilizando buffer de N registros de escrituras: 4
    Mientras haya escrituras en prueba.dat hacer 5
        Leer una escritura.
        Si la escritura es v3lida6 entonces
            Si es la primera escritura validada entonces
                Inicializar los registros significativos con los datos de esa escritura.
                //ya ser3 la de menor posici3n puesto que hacemos un barrido secuencial
            sino
                Comparar n3 de escritura (para obtener primera y 3ltima) y actualizarlas si es preciso
            fsi
            Incrementar contador escrituras validadas.
        fmientras
        Obtener la escritura de la 3ltima posici3n.
        A3adir la informaci3n del struct info al fichero informe.txt por el final.
    fpara
Desmontar el dispositivo virtual.
```

² **Mejora:** Las entradas tambi3n las pod3is haber le3do todas de golpe previamente al inicio del bucle, con una sola llamada a `mi_read()` utilizando un buffer del tama3o `NUMPROCESOS * sizeof(struct entrada)` o llamando a vuestra funci3n `mi_dir()` en su versi3n simple. Entonces en este paso lo que habr3a que hacer es leer la entrada pero del buffer, sin necesidad de acceder para cada una al dispositivo.

³ Pod3is utilizar la funci3n `strchr()` con el car3cter '_' para obtener los caracteres del PID, y luego pasarlos a entero con la funci3n `atoi()`. El registro `info` es de tipo `struct INFORMACION`.

⁴ La cantidad de registros, multiplicado por el `sizeof(struct REGISTRO)` mejor si es un m3ltiplo de `BLOCKSIZE`. En una plataforma de 64 bits el tama3o del `struct REGISTRO` utilizando el tipo `time_t` para la fecha, es 24 bytes, entonces N podr3a ser por ejemplo 256 que multiplicado por 24 da 6144, que equivale a 6 bloques de 1024Bs.

```
int cant_registros_buffer_escrituras = 256;
struct REGISTRO buffer_escrituras[cant_registros_buffer_escrituras];
memset(buffer_escrituras, 0, sizeof(buffer_escrituras));
while (mi_read(prueba, buffer_escrituras, offset, sizeof(buffer_escrituras)) > 0) {...}
```

Hay que limpiar el buffer de lectura antes de cada nuevo uso!!!

⁵ Se ahorrar3 tiempo de ejecuci3n teniendo una **cach3 de directorios** para realizar el `mi_read()` y no tener que llamar a `buscar_entrada()` para leer cada escritura de un mismo proceso (mismo camino).

Tambi3n si en vez de leer registro a registro del dispositivo, se explora en un buffer en memoria principal los registros que caben en un bloque, o en varios.

⁶ Para saberlo verificaremos que el campo `pid` de la escritura coincida con el del proceso, ya que podr3a haber basura debido a que tratamos con ficheros dispersos y hemos escrito en direcciones aleatorias.

Aunque las fechas se guarden en formato epoch se han de mostrar por pantalla en formato legible.

Podéis utilizar la función `strftime()`, que utilizábamos para ver los sellos de tiempo del inodo, y que podemos formatear a nuestro gusto. Por ejemplo usando el siguiente argumento `"%a %d-%m-%Y %H:%M:%S"` con `strftime()` se produciría el siguiente formato:

Fri 25-05-2018 11:44:47

Otra forma más simple pero más rígida es utilizar la función `asctime()`, por ejemplo:

`asctime(localtime(&info.PrimerEscritura.fecha))`

que mostrará la fecha con el siguiente formato:

Fri May 25 11:44:47 2018

Hay que mostrar por pantalla cuánto os tarda la verificación de escrituras. Podéis hacerlo desde consola mediante el comando `time`.

EJECUCIÓN DE LA VERIFICACIÓN DE LA SIMULACION DE EJEMPLO DE LA ETAPA ANTERIOR

```
$ time ./verificacion disco /simul_20210524105645/
dir_sim: /simul_20210524105645/
numentradas: 100 NUMPROCESOS: 100
[1] 50 escrituras validadas en /simul_20210524105645/proceso_10982/prueba.dat]
[2] 50 escrituras validadas en /simul_20210524105645/proceso_10983/prueba.dat]
[3] 50 escrituras validadas en /simul_20210524105645/proceso_10984/prueba.dat]
[4] 50 escrituras validadas en /simul_20210524105645/proceso_10985/prueba.dat]
[5] 50 escrituras validadas en /simul_20210524105645/proceso_10986/prueba.dat]
[6] 49 escrituras validadas en /simul_20210524105645/proceso_10987/prueba.dat]
[7] 50 escrituras validadas en /simul_20210524105645/proceso_10988/prueba.dat]
[8] 50 escrituras validadas en /simul_20210524105645/proceso_10989/prueba.dat]
[9] 50 escrituras validadas en /simul_20210524105645/proceso_10990/prueba.dat]
[10] 50 escrituras validadas en /simul_20210524105645/proceso_10991/prueba.dat]
[11] 50 escrituras validadas en /simul_20210524105645/proceso_10992/prueba.dat]
[12] 50 escrituras validadas en /simul_20210524105645/proceso_10993/prueba.dat]
[13] 50 escrituras validadas en /simul_20210524105645/proceso_10994/prueba.dat]
[14] 50 escrituras validadas en /simul_20210524105645/proceso_10995/prueba.dat]
[15] 50 escrituras validadas en /simul_20210524105645/proceso_10999/prueba.dat]
[16] 50 escrituras validadas en /simul_20210524105645/proceso_11000/prueba.dat]
[17] 50 escrituras validadas en /simul_20210524105645/proceso_11001/prueba.dat]
[18] 50 escrituras validadas en /simul_20210524105645/proceso_11003/prueba.dat]
```



```
[64) 50 escrituras validadas en /simul_20210524105645/proceso_11065/prueba.dat]
[65) 50 escrituras validadas en /simul_20210524105645/proceso_11066/prueba.dat]
[66) 50 escrituras validadas en /simul_20210524105645/proceso_11067/prueba.dat]
[67) 50 escrituras validadas en /simul_20210524105645/proceso_11070/prueba.dat]
[68) 50 escrituras validadas en /simul_20210524105645/proceso_11071/prueba.dat]
[69) 50 escrituras validadas en /simul_20210524105645/proceso_11072/prueba.dat]
[70) 50 escrituras validadas en /simul_20210524105645/proceso_11073/prueba.dat]
[71) 50 escrituras validadas en /simul_20210524105645/proceso_11074/prueba.dat]
[72) 50 escrituras validadas en /simul_20210524105645/proceso_11075/prueba.dat]
[73) 50 escrituras validadas en /simul_20210524105645/proceso_11078/prueba.dat]
[74) 50 escrituras validadas en /simul_20210524105645/proceso_11079/prueba.dat]
[75) 50 escrituras validadas en /simul_20210524105645/proceso_11080/prueba.dat]
[76) 50 escrituras validadas en /simul_20210524105645/proceso_11082/prueba.dat]
[77) 50 escrituras validadas en /simul_20210524105645/proceso_11083/prueba.dat]
[78) 50 escrituras validadas en /simul_20210524105645/proceso_11081/prueba.dat]
[79) 50 escrituras validadas en /simul_20210524105645/proceso_11084/prueba.dat]
[80) 50 escrituras validadas en /simul_20210524105645/proceso_11085/prueba.dat]
[81) 50 escrituras validadas en /simul_20210524105645/proceso_11086/prueba.dat]
[82) 50 escrituras validadas en /simul_20210524105645/proceso_11089/prueba.dat]
[83) 50 escrituras validadas en /simul_20210524105645/proceso_11090/prueba.dat]
[84) 50 escrituras validadas en /simul_20210524105645/proceso_11091/prueba.dat]
[85) 50 escrituras validadas en /simul_20210524105645/proceso_11092/prueba.dat]
[86) 50 escrituras validadas en /simul_20210524105645/proceso_11093/prueba.dat]
[87) 50 escrituras validadas en /simul_20210524105645/proceso_11094/prueba.dat]
[88) 50 escrituras validadas en /simul_20210524105645/proceso_11095/prueba.dat]
[89) 50 escrituras validadas en /simul_20210524105645/proceso_11096/prueba.dat]
[90) 50 escrituras validadas en /simul_20210524105645/proceso_11097/prueba.dat]
[91) 50 escrituras validadas en /simul_20210524105645/proceso_11100/prueba.dat]
[92) 50 escrituras validadas en /simul_20210524105645/proceso_11101/prueba.dat]
[93) 50 escrituras validadas en /simul_20210524105645/proceso_11102/prueba.dat]
[94) 50 escrituras validadas en /simul_20210524105645/proceso_11103/prueba.dat]
[95) 50 escrituras validadas en /simul_20210524105645/proceso_11104/prueba.dat]
[96) 50 escrituras validadas en /simul_20210524105645/proceso_11105/prueba.dat]
[97) 50 escrituras validadas en /simul_20210524105645/proceso_11106/prueba.dat]
[98) 50 escrituras validadas en /simul_20210524105645/proceso_11109/prueba.dat]
[99) 50 escrituras validadas en /simul_20210524105645/proceso_11110/prueba.dat]
[100) 50 escrituras validadas en /simul_20210524105645/proceso_11111/prueba.dat]
```

```
real    0m7,977s
user    0m4,263s
sys     0m3,704s
```

```
$ ./mi_cat disco /simul_20210524105645/informe.txt > resultado.txt
```

```
Total_leidos 25550
```


\$ ls -l resultado.txt

-rw-rw-r-- 1 uib uib 25550 de maig 24 11:11 resultado.txt

\$ cat resultado.txt

PID: 10982

Numero de escrituras: 50

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 154409 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 247776 | Thu May 26 18:34:22 2022 |
| Menor Posición | 30 | 6070 | Thu May 26 18:34:21 2022 |
| Mayor Posición | 31 | 479665 | Thu May 26 18:34:21 2022 |

PID: 10983

Numero de escrituras: 50

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 452816 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 246650 | Thu May 26 18:34:22 2022 |
| Menor Posición | 19 | 8981 | Thu May 26 18:34:21 2022 |
| Mayor Posición | 13 | 492815 | Thu May 26 18:34:20 2022 |

PID: 10984

Numero de escrituras: 50

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 369931 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 30400 | Thu May 26 18:34:22 2022 |
| Menor Posición | 26 | 205 | Thu May 26 18:34:21 2022 |
| Mayor Posición | 11 | 496854 | Thu May 26 18:34:20 2022 |

PID: 10985

Numero de escrituras: 50

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 314741 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 422224 | Thu May 26 18:34:22 2022 |
| Menor Posición | 10 | 19542 | Thu May 26 18:34:20 2022 |
| Mayor Posición | 46 | 496710 | Thu May 26 18:34:22 2022 |

PID: 10986

Numero de escrituras: 50

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 491736 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 353855 | Thu May 26 18:34:23 2022 |
| Menor Posición | 31 | 1771 | Thu May 26 18:34:22 2022 |
| Mayor Posición | 15 | 499326 | Thu May 26 18:34:21 2022 |

PID: 10987

Numero de escrituras: 49

| | | | |
|-------------------|----|--------|--------------------------|
| Primera Escritura | 1 | 476061 | Thu May 26 18:34:20 2022 |
| Ultima Escritura | 50 | 170138 | Thu May 26 18:34:23 2022 |
| Menor Posición | 17 | 1117 | Thu May 26 18:34:21 2022 |
| Mayor Posición | 42 | 487747 | Thu May 26 18:34:22 2022 |

```
PID: 10988
Numero de escrituras: 50
Primera Escritura    1      234335    Thu May 26 18:34:20 2022
Ultima Escritura     50      84542    Thu May 26 18:34:23 2022
Menor Posición       32      843      Thu May 26 18:34:22 2022
Mayor Posición       17      484807   Thu May 26 18:34:21 2022
```

...

\$./leer_sf disco

```
DATOS DEL SUPERBLOQUE
posPrimerBloqueMB = 1
posUltimoBloqueMB = 13
posPrimerBloqueAI = 14
posUltimoBloqueAI = 3138
posPrimerBloqueDatos = 3139
posUltimoBloqueDatos = 99999
posInodoRaiz = 0
posPrimerInodoLibre = 203
cantBloquesLibres = 88534 7
cantInodosLibres = 24797 8
totBloques = 100000
totInodos = 25000
```

Observaciones:

- El nº de escrituras validadas ha de ser por lo general 50, salvo que por casualidad dos hayan ido a para a la misma posición cuando se ha generado el nº de registro aleatorio, y una se haya sobrescrito a la otra.
- La primera escritura será la que lleva por nº el 1 (y tendrá el sello de tiempo menor), y la última la que lleva por nº el 50 (y tendrá el sello de tiempo mayor), salvo que haya alguna sobrescritura.
- Las posiciones estarán en el rango [0, 499.999].

⁷ Dependerá en cada ejecución de las ubicaciones aleatorias de las escrituras (diferentes niveles de bloques de punteros o ninguno)

⁸ Se han reservado 202 inodos: 1 para el directorio de simulación + 100 directorios de procesos + 100 ficheros prueba.dat + 1 fichero informe.txt

- Opcionalmente los sellos de tiempo se pueden guardar en **microsegundos**, usando la función `gettimeofday()` y el `struct timeval` de la librería `sys/time.h`.