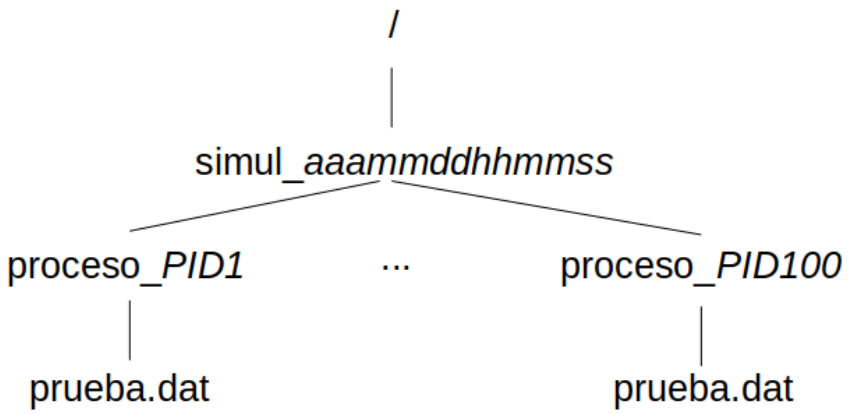
Nivel 12: **simulacion.c**

En este nivel se trata de crear un programa **simulador** (simulacion.c y su correspondiente simulacion.h), encargado de crear unos **procesos** de prueba que accedan de forma **concurrente** al sistema de ficheros, de modo que se pueda comprobar el correcto funcionamiento de nuestra biblioteca de funciones para más de un proceso en ejecución.

Primeramente se creará el directorio llamado `simul_aaaammddhhmmss` (en el directorio raíz) donde *aaaa* es el año, *mm* es el mes, *dd* es el día, *hh* es la hora, *mm* es el minuto y *ss* es el segundo de creación.
Se han de generar **100 procesos** de prueba[1] **cada 0,2 segundos**. Cada proceso creará un directorio llamado `proceso_n` dentro del directorio `simul_aaaammddhhmmss`, donde n es el **PID** del proceso. Además, dentro del directorio "`proceso_n`", cada proceso creará un fichero denominado "`prueba.dat`".



Cada 0,05 segundos y un total de **50 veces**, cada proceso escribirá dentro del fichero "`prueba.dat`" un registro del siguiente tipo, declarado en **simulacion.h**:

```
struct REGISTRO { //sizeof(struct REGISTRO): 24
   time_t fecha; //Precisión segundos
   pid_t pid; //PID del proceso que lo ha creado
   int nEscritura; //Entero con el número de escritura (de 1 a 50)
   int nRegistro; //Entero con el número del registro dentro del fichero
};
```

Un algoritmo más detallado a seguir sería el siguiente:

[1] Recordad que la función fork() permite crear un nuevo proceso

```
Asociar la señal SIGCHLD al enterrador[2]
Comprobar la sintaxis  // uso: ./simulacion <disco>
Montar el dispositivo //padre
Crear el directorio de simulación: /simul_aaaammddhhmmss
Para proceso:=1 hasta proceso<=NUMPROCESOS hacer
   pid:=fork()
   Si es el hijo entonces //pid = 0
     Montar el dispositivo //hijo[3]
        Crear el directorio del proceso añadiendo el PID al nombre
        Crear el fichero prueba.dat
        Inicializar la semilla de números aleatorios[4]: srand(time(NULL) + getpid());
        Para nescritura:=1 hasta nescritura<=NUMESCRITURAS hacer
          Inicializar el registro:
            registro.fecha = time(NULL)
            registro.pid = getpid()
            registro.nEscritura = nescritura
            registro.nRegistro = rand() % REGMAX[5]
          Escribir el registro con mi_write() en registro.nRegistro * sizeof(struct REGISTRO)
          Esperar 0,05 seg para hacer la siguiente escritura
        fpara
        Desmontar el dispositivo //hijo
        exit(0) //Necesario para que se emita la señal SIGCHLD
    fsi
    Esperar 0,2 seg para lanzar siguiente proceso
fpara

 //Permitir que el padre espere por todos los hijos[6]:
 Mientras acabados < NUMPROCESOS hacer
    pause();
 fmientras

Desmontar el dispositivo //padre
```

[2] Recordemos que cuando un proceso hijo termina, el sistema guarda el **PID** (Identificador) y su **estado** (un parámetro) para dárselo a su padre. Hasta entonces el proceso finalizado entra en estado **zombie**. Cuando un proceso finaliza, toda la memoria y recursos asociados con dicho proceso son liberados, pero la entrada del mismo en la tabla de procesos aún existe, para cuando su padre llame a la función wait() devolverle su PID y estado.

[3] El proceso principal hace el bmount() por lo que abre el fichero con el open(). Luego hace varios fork()s, y los hijos comparten los datos de los descriptores del fichero del padre. También se comparte el puntero (cambiado por el lseek()). Un proceso hace el lseek() para leer o escribir, otro hace lo mismo y lo deja cambiado, cuando el primero hace su operación de lectura o escritura accede a posiciones diferentes al lseek que hizo previamente. Hacer el bmount() nuevamente elimina ese problema de concurrencia.

[4] Para que genere números diferentes en cada ejecución para obtener los números de registro.

[5] Nuestro sistema de ficheros nos permitiría un REGMAX = (((12+256+256²+256³)-1)*BLOCKSIZE) /sizeof(struct registro) pero en la simulación lo limitaremos a 500.000 registros (valor de REGMAX)

[6] Tendremos una variable global, acabados, inicializada a 0 para llevar la cuenta del nº de procesos finalizados, y que el **enterrador** irá incrementado

```
exit(0) // o return 0[7]
```

Función **enterrador**:

```
void reaper(){
  pid_t ended;
  signal(SIGCHLD, reaper);
  while ((ended=waitpid(-1, NULL, WNOHANG))>0) {
     acabados++;
  }
}
```

Observaciones:

- Se precisan las siguientes cabeceras:

```
#include <sys/wait.h>
#include <signal.h>
```

- Dado que en cada proceso hijo montamos y desmontamos el dispositivo virtual, tendremos que modificar las funciones bmount() y bumount() de **bloques.c**:

```
int bmount(const char *camino) {
  if (descriptor > 0) {
     close(descriptor);
  }
 ...
}

int bumount() {
  descriptor = close(descriptor);
...
}
```

Se puede provocar una espera mediante la función usleep(), especificando el tiempo en **microsegundos**. Hay que tener en cuenta que esta función es cancelada por la llegada de alguna señal (por ejemplo cuando un hijo acaba y se envía la señal SIGCHLD), y por tanto la ejecución de la simulación duraría menos de 100*0,2 segundos. Alternativamente se puede usar nanosleep(), especificando el tiempo en **nanosegundos**, y forzar la espera en caso de que llegue una señal de la siguiente manera:

```
void my_sleep(unsigned msec) { //tiempo en milisegundos
  struct timespec req, rem;
  int err;
  req.tv_sec = msec / 1000;
  req.tv_nsec = (msec % 1000) * 1000000;
  while ((req.tv_sec != 0) || (req.tv_nsec != 0)) {
     if (nanosleep(&req, &rem) == 0)
```

[7] Hay discusiones a favor y en contra de la conveniencia en C de usar exit() o return para finalizar el main():
https://stackoverflow.com/questions/461449/return-statement-vs-exit-in-main
https://stackoverflow.com/questions/3463551/what-is-the-difference-between-exit-and-return/3463562

```
            break;
        err = errno;
        // Interrupted; continue
        if (err == EINTR) {
            req.tv_sec = rem.tv_sec;
            req.tv_nsec = rem.tv_nsec;
        }
    }
}
```

Se aconseja borrar el dispositivo virtual antes de crear otro con el mismo nombre y empezar las pruebas con 1 solo proceso y 5-10 escrituras, luego ir aumentando el nº de procesos y nº de escrituras e ir comprobando los resultados poco a poco antes de expandir la simulación a los 100 procesos y 50 escrituras.

Ejemplo de testing:

```
*** Simulación de 3 procesos realizando cada uno 10 escrituras ***
Directorio simulación: /simul_20180516143353/
[simulación.c → Escritura 1 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 2 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 3 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 4 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 1 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 5 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 2 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 6 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 7 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 3 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 4 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 8 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 1 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 9 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 5 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 2 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 10 en /simul_20180516143353/proceso_31653/prueba.dat]
[simulación.c → Escritura 6 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 3 en /simul_20180516143353/proceso_31656/prueba.dat]
Proceso 1: Completadas 10 escrituras en
/simul_20180516143353/proceso_31653/prueba.dat
[simulación.c → Escritura 7 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Acabado proceso con PID 31653, total acabados: 1
[simulación.c → acabados: 1
[simulación.c → Escritura 4 en /simul_20180516143353/proceso_31656/prueba.dat]
```

```
[simulación.c → Escritura 8 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 5 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 9 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 6 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 10 en /simul_20180516143353/proceso_31655/prueba.dat]
[simulación.c → Escritura 7 en /simul_20180516143353/proceso_31656/prueba.dat]
Proceso 2: Completadas 10 escrituras en
/simul_20180516143353/proceso_31655/prueba.dat
[simulación.c → Acabado proceso con PID 31655, total acabados: 2
[simulación.c → acabados: 2
[simulación.c → Escritura 8 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 9 en /simul_20180516143353/proceso_31656/prueba.dat]
[simulación.c → Escritura 10 en /simul_20180516143353/proceso_31656/prueba.dat]
Proceso 3: Completadas 10 escrituras en
/simul_20180516143353/proceso_31656/prueba.dat
[simulación.c → Acabado proceso con PID 31656, total acabados: 3
Total de procesos terminados: 3.
```

Una vez testeado el funcionamiento para unos pocos procesos y unas pocas escrituras podéis eliminar las impresiones entre corchetes y lanzarlo, para los 100 procesos con 50 escrituras cada una, con el comando time [8] delante para contrastar vuestro tiempo de ejecución. **Es imprescindible la impresión del nombre del directorio de simulación para la posterior verificación de las escrituras ya que lo necesitaremos como parámetro.**

```
$ ./mi_mkfs disco 100000
$ /usr/bin/time ./simulacion disco  #mostrar dir_simulacion
*** Simulación de 100 procesos realizando cada uno 50 escrituras ***
Directorio simulación: /simul_20180523133828/
Proceso 1: Completadas 50 escrituras en /simul_20180523133828/proceso_16776/prueba.dat
Proceso 2: Completadas 50 escrituras en /simul_20180523133828/proceso_16779/prueba.dat
Proceso 3: Completadas 50 escrituras en /simul_20180523133828/proceso_16780/prueba.dat
Proceso 4: Completadas 50 escrituras en /simul_20180523133828/proceso_16782/prueba.dat
Proceso 5: Completadas 50 escrituras en /simul_20180523133828/proceso_16783/prueba.dat
Proceso 6: Completadas 50 escrituras en /simul_20180523133828/proceso_16784/prueba.dat
Proceso 7: Completadas 50 escrituras en /simul_20180523133828/proceso_16787/prueba.dat
```

[8] Time muestra 3 tiempos:

- el **real** de ejecución: es el tiempo total transcurrido desde que ha invocado el comando (incluyendo intervalos de tiempo utilizados por otros procesos y tiempos de espera para E/S). Se le denomina a veces como tiempo de reloj, porque es tiempo que ha transcurrido en nuestro reloj.
- el de **usuario:** es sólo el tiempo de CPU utilizado en la ejecución del proceso.
- el de **sistema:** es la cantidad de tiempo consumido en el kernel, que es el tiempo empleado en contestar peticiones del sistema.

```
Proceso 8: Completadas 50 escrituras en /simul_20180523133828/proceso_16788/prueba.dat
Proceso 9: Completadas 50 escrituras en /simul_20180523133828/proceso_16790/prueba.dat
Proceso 10: Completadas 50 escrituras en /simul_20180523133828/proceso_16791/prueba.dat
Proceso 11: Completadas 50 escrituras en /simul_20180523133828/proceso_16792/prueba.dat
Proceso 12: Completadas 50 escrituras en /simul_20180523133828/proceso_16795/prueba.dat
Proceso 13: Completadas 50 escrituras en /simul_20180523133828/proceso_16796/prueba.dat
Proceso 14: Completadas 50 escrituras en /simul_20180523133828/proceso_16797/prueba.dat
Proceso 16: Completadas 50 escrituras en /simul_20180523133828/proceso_16800/prueba.dat
Proceso 15: Completadas 50 escrituras en /simul_20180523133828/proceso_16799/prueba.dat
Proceso 17: Completadas 50 escrituras en /simul_20180523133828/proceso_16801/prueba.dat
Proceso 18: Completadas 50 escrituras en /simul_20180523133828/proceso_16803/prueba.dat
Proceso 19: Completadas 50 escrituras en /simul_20180523133828/proceso_16805/prueba.dat
Proceso 20: Completadas 50 escrituras en /simul_20180523133828/proceso_16806/prueba.dat
Proceso 21: Completadas 50 escrituras en /simul_20180523133828/proceso_16807/prueba.dat
Proceso 22: Completadas 50 escrituras en /simul_20180523133828/proceso_16810/prueba.dat
Proceso 23: Completadas 50 escrituras en /simul_20180523133828/proceso_16811/prueba.dat
Proceso 24: Completadas 50 escrituras en /simul_20180523133828/proceso_16814/prueba.dat
Proceso 25: Completadas 50 escrituras en /simul_20180523133828/proceso_16816/prueba.dat
Proceso 26: Completadas 50 escrituras en /simul_20180523133828/proceso_16817/prueba.dat
Proceso 27: Completadas 50 escrituras en /simul_20180523133828/proceso_16818/prueba.dat
Proceso 29: Completadas 50 escrituras en /simul_20180523133828/proceso_16821/prueba.dat
Proceso 28: Completadas 50 escrituras en /simul_20180523133828/proceso_16820/prueba.dat
Proceso 31: Completadas 50 escrituras en /simul_20180523133828/proceso_16823/prueba.dat
Proceso 30: Completadas 50 escrituras en /simul_20180523133828/proceso_16822/prueba.dat
Proceso 32: Completadas 50 escrituras en /simul_20180523133828/proceso_16824/prueba.dat
Proceso 33: Completadas 50 escrituras en /simul_20180523133828/proceso_16827/prueba.dat
Proceso 34: Completadas 50 escrituras en /simul_20180523133828/proceso_16828/prueba.dat
Proceso 35: Completadas 50 escrituras en /simul_20180523133828/proceso_16829/prueba.dat
Proceso 36: Completadas 50 escrituras en /simul_20180523133828/proceso_16831/prueba.dat
Proceso 37: Completadas 50 escrituras en /simul_20180523133828/proceso_16832/prueba.dat
Proceso 38: Completadas 50 escrituras en /simul_20180523133828/proceso_16833/prueba.dat
Proceso 39: Completadas 50 escrituras en /simul_20180523133828/proceso_16834/prueba.dat
Proceso 40: Completadas 50 escrituras en /simul_20180523133828/proceso_16836/prueba.dat
Proceso 41: Completadas 50 escrituras en /simul_20180523133828/proceso_16839/prueba.dat
Proceso 42: Completadas 50 escrituras en /simul_20180523133828/proceso_16840/prueba.dat
Proceso 44: Completadas 50 escrituras en /simul_20180523133828/proceso_16843/prueba.dat
Proceso 43: Completadas 50 escrituras en /simul_20180523133828/proceso_16842/prueba.dat
Proceso 45: Completadas 50 escrituras en /simul_20180523133828/proceso_16844/prueba.dat
Proceso 46: Completadas 50 escrituras en /simul_20180523133828/proceso_16845/prueba.dat
Proceso 47: Completadas 50 escrituras en /simul_20180523133828/proceso_16846/prueba.dat
Proceso 48: Completadas 50 escrituras en /simul_20180523133828/proceso_16847/prueba.dat
Proceso 50: Completadas 50 escrituras en /simul_20180523133828/proceso_16850/prueba.dat
Proceso 49: Completadas 50 escrituras en /simul_20180523133828/proceso_16849/prueba.dat
Proceso 52: Completadas 50 escrituras en /simul_20180523133828/proceso_16852/prueba.dat
Proceso 51: Completadas 50 escrituras en /simul_20180523133828/proceso_16851/prueba.dat
Proceso 53: Completadas 50 escrituras en /simul_20180523133828/proceso_16853/prueba.dat
```

```
Proceso 55: Completadas 50 escrituras en /simul_20180523133828/proceso_16856/prueba.dat
Proceso 56: Completadas 50 escrituras en /simul_20180523133828/proceso_16857/prueba.dat
Proceso 54: Completadas 50 escrituras en /simul_20180523133828/proceso_16855/prueba.dat
Proceso 57: Completadas 50 escrituras en /simul_20180523133828/proceso_16858/prueba.dat
Proceso 58: Completadas 50 escrituras en /simul_20180523133828/proceso_16860/prueba.dat
Proceso 59: Completadas 50 escrituras en /simul_20180523133828/proceso_16862/prueba.dat
Proceso 60: Completadas 50 escrituras en /simul_20180523133828/proceso_16863/prueba.dat
Proceso 61: Completadas 50 escrituras en /simul_20180523133828/proceso_16864/prueba.dat
Proceso 62: Completadas 50 escrituras en /simul_20180523133828/proceso_16865/prueba.dat
Proceso 63: Completadas 50 escrituras en /simul_20180523133828/proceso_16867/prueba.dat
Proceso 64: Completadas 50 escrituras en /simul_20180523133828/proceso_16868/prueba.dat
Proceso 65: Completadas 50 escrituras en /simul_20180523133828/proceso_16870/prueba.dat
Proceso 66: Completadas 50 escrituras en /simul_20180523133828/proceso_16872/prueba.dat
Proceso 67: Completadas 50 escrituras en /simul_20180523133828/proceso_16873/prueba.dat
Proceso 68: Completadas 50 escrituras en /simul_20180523133828/proceso_16874/prueba.dat
Proceso 69: Completadas 50 escrituras en /simul_20180523133828/proceso_16875/prueba.dat
Proceso 70: Completadas 50 escrituras en /simul_20180523133828/proceso_16876/prueba.dat
Proceso 71: Completadas 50 escrituras en /simul_20180523133828/proceso_16877/prueba.dat
Proceso 72: Completadas 50 escrituras en /simul_20180523133828/proceso_16879/prueba.dat
Proceso 73: Completadas 50 escrituras en /simul_20180523133828/proceso_16880/prueba.dat
Proceso 74: Completadas 50 escrituras en /simul_20180523133828/proceso_16881/prueba.dat
Proceso 75: Completadas 50 escrituras en /simul_20180523133828/proceso_16882/prueba.dat
Proceso 76: Completadas 50 escrituras en /simul_20180523133828/proceso_16883/prueba.dat
Proceso 77: Completadas 50 escrituras en /simul_20180523133828/proceso_16885/prueba.dat
Proceso 80: Completadas 50 escrituras en /simul_20180523133828/proceso_16888/prueba.dat
Proceso 78: Completadas 50 escrituras en /simul_20180523133828/proceso_16886/prueba.dat
Proceso 79: Completadas 50 escrituras en /simul_20180523133828/proceso_16887/prueba.dat
Proceso 81: Completadas 50 escrituras en /simul_20180523133828/proceso_16889/prueba.dat
Proceso 82: Completadas 50 escrituras en /simul_20180523133828/proceso_16890/prueba.dat
Proceso 83: Completadas 50 escrituras en /simul_20180523133828/proceso_16892/prueba.dat
Proceso 84: Completadas 50 escrituras en /simul_20180523133828/proceso_16893/prueba.dat
Proceso 85: Completadas 50 escrituras en /simul_20180523133828/proceso_16894/prueba.dat
Proceso 86: Completadas 50 escrituras en /simul_20180523133828/proceso_16895/prueba.dat
Proceso 88: Completadas 50 escrituras en /simul_20180523133828/proceso_16897/prueba.dat
Proceso 87: Completadas 50 escrituras en /simul_20180523133828/proceso_16896/prueba.dat
Proceso 89: Completadas 50 escrituras en /simul_20180523133828/proceso_16899/prueba.dat
Proceso 90: Completadas 50 escrituras en /simul_20180523133828/proceso_16901/prueba.dat
Proceso 91: Completadas 50 escrituras en /simul_20180523133828/proceso_16902/prueba.dat
Proceso 92: Completadas 50 escrituras en /simul_20180523133828/proceso_16904/prueba.dat
Proceso 93: Completadas 50 escrituras en /simul_20180523133828/proceso_16905/prueba.dat
Proceso 94: Completadas 50 escrituras en /simul_20180523133828/proceso_16906/prueba.dat
Proceso 95: Completadas 50 escrituras en /simul_20180523133828/proceso_16907/prueba.dat
Proceso 96: Completadas 50 escrituras en /simul_20180523133828/proceso_16908/prueba.dat
Proceso 97: Completadas 50 escrituras en /simul_20180523133828/proceso_16910/prueba.dat
Proceso 99: Completadas 50 escrituras en /simul_20180523133828/proceso_16912/prueba.dat
Proceso 98: Completadas 50 escrituras en /simul_20180523133828/proceso_16911/prueba.dat
```

```
Proceso 100: Completadas 50 escrituras en
/simul_20180523133828/proceso_16914/prueba.dat
Total de procesos terminados: 100.

real    0m14.197s
user    0m0.239s
sys     0m0.745s
```

Observación: Un proceso puede acabar antes que otro iniciado antes que él