

Practica 3: Libckpt

- Implementar un programa lanzador en 'c' que tome como parámetros los nombres de otros programas que se tienen que ejecutar (cont).
- Mediante el uso de fork(), de execlp() y de wait(), el programa ejecutará estos programas y detectará cuando alguno de ellos termina de forma anormal para volverlos a ejecutar.

- execlp(nombre,nombre,NULL)
- fin = wait(&status);
if (WIFEXITED(status)) ...

Practica 3: Libckpt

- En una segunda parte, los programas que se ejecutan utilizarán la librería **libckpt** para establecer puntos de recuperación, de forma que cuando sean relanzados no empiecen de nuevo, sino desde el último checkpoint establecido.

```
cont: cont.c cmain.o libckpt.a checkpoint.h
$(CC) $(CFLAGS) -o cont cmain.o cont.c libckpt.a
```

```
bash> ./cont =recover
```

```
#include "checkpoint.h"
int ckpt_target(int argc, char **argv) {
    checkpoint_here();
}
```

Practica 3: Libckpt

- En una tercera parte, se utilizará el fichero de configuración de libckpt (.ckptrc) para modificar la forma en que la librería realiza los puntos de recuperación.

```
checkpointing on
directory .
verbose off
incremental on
maxfiles 5
fork off
maxtime 0
```