



# Universidad Católica Argentina

Sistemas Operativos  
Lic. Ing. Osvaldo Clúa

Laboratorio: System calls,  
procesos y threads

# System calls y library calls

- Compilar en Linux el siguiente programa:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

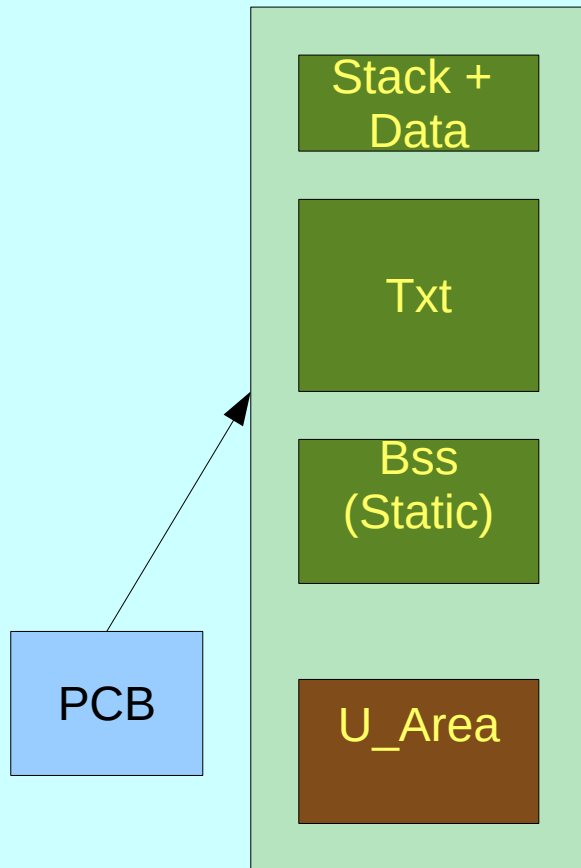
int main(){
    pid_t yo;
    yo=getpid();
    printf ("Yo soy el proceso %d\n",yo);
}
```

- Verificar los system calls con strace
- Verificar los library calls con ltrace
- Probar la opción -c de ambos comandos

# Código Linux

- Ud encontrará un archivo zip con los subdirectorios
  - env donde hay ejemplos de acceso al environment
  - fork
  - thread
  - corutinas

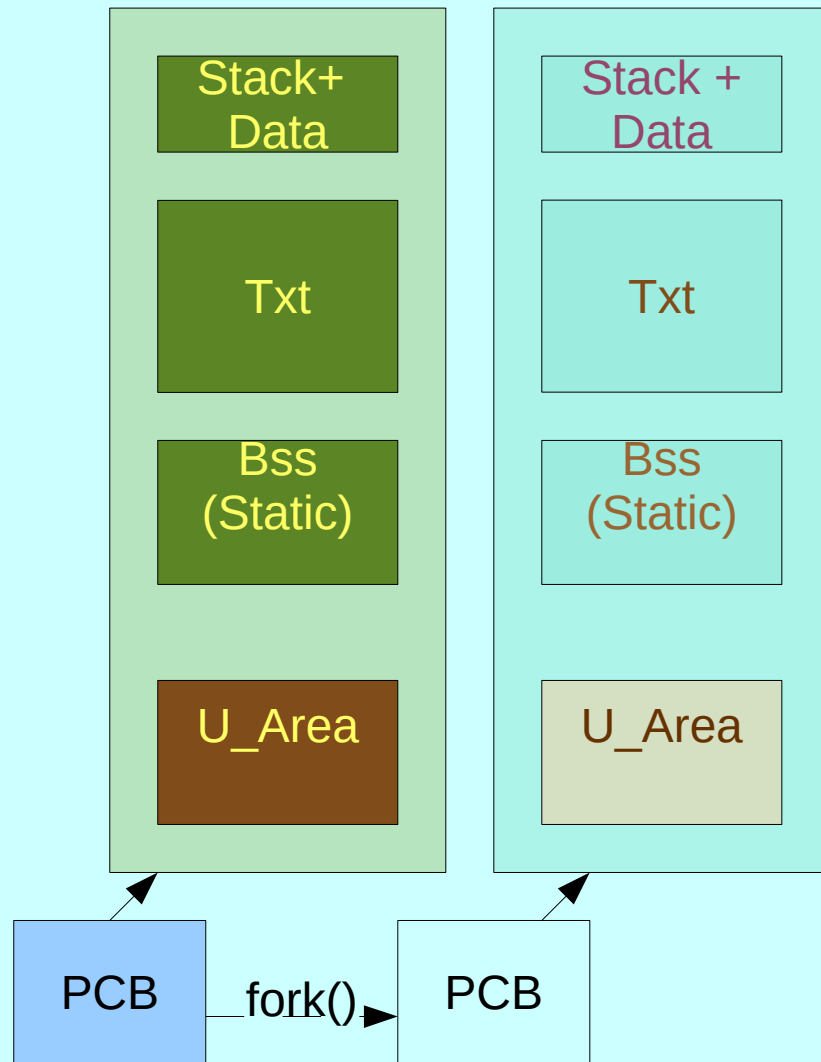
# Directorio env



Los programas muestran dos formas distintas de acceder tanto a los argumentos como a las variables de ambiente.

Ubique estas formas en un diagrama como el adjunto.

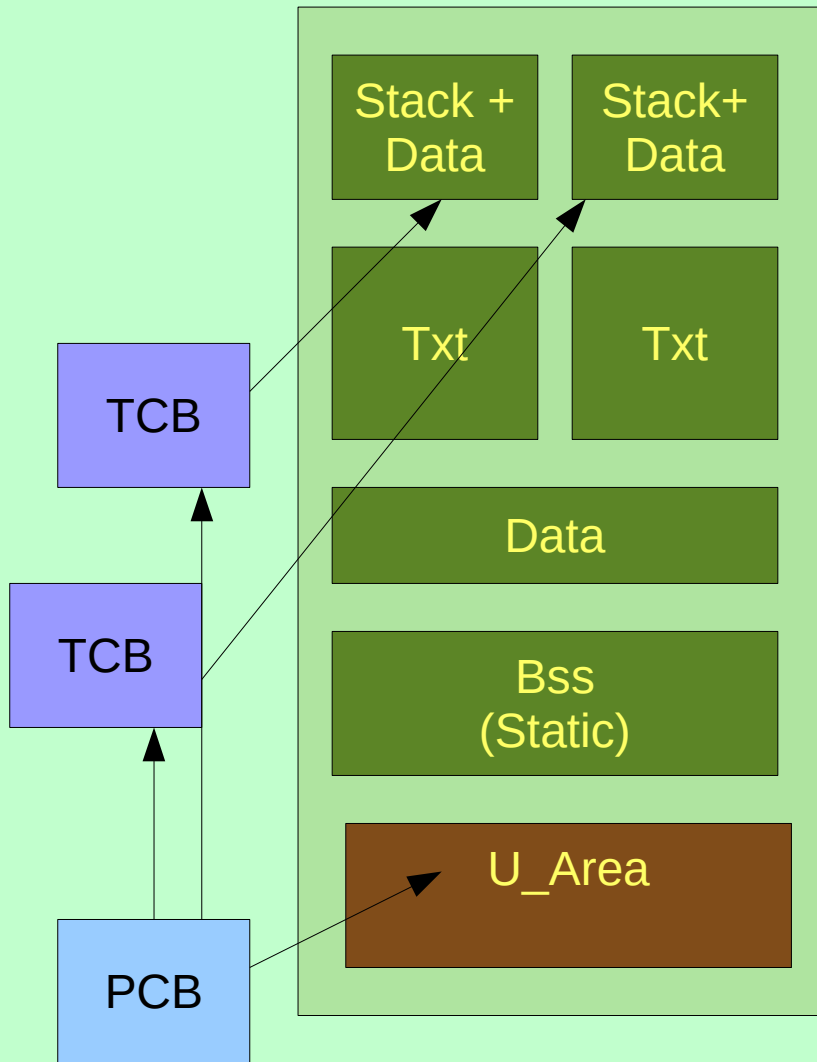
# Directorio fork



Los programas demuestran el uso de `fork()` y `exec()`.

Indique la evolución de estos en el gráfico adjunto.

# Threads

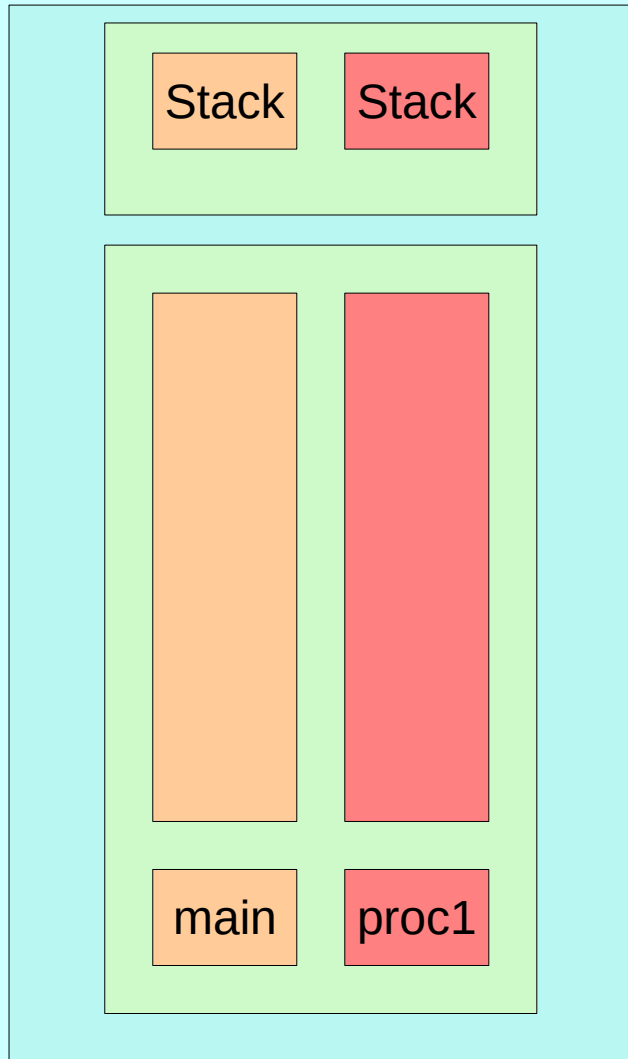


Demuestra el uso de threads y su diferencia con fork.

El ejemplo de Perl requiere una versión de Perl compilada con threads.

Use strace para **perfilar** los programas (¿porque no usar gprof(1)?).

# Corutinas



- Indicar en el diagrama con números de instrucción y flechas los cambios de contexto entre programa y subrutina.
- ¿Cuántos procesos hay?
- Reescriba con `setjmp (3)` y `longjmp (3)`.