**Create Archives**

ar cr libtest.a test1.o test2.o

**Shared Libraries**

gcc -c -fPIC test1.c

PIC stands for position-independent code.

Then you combine the object files into a shared library, like this:

gcc -shared -fPIC -o libtest.so test1.o test2.o

**Using *LD\_LIBRARY\_PATH***

gcc -o app app.o -L. -ltest -Wl,-rpath,/usr/local/lib

**Dynamic Loading and Unloading**

dlopen (“libtest.so”, RTLD\_LAZY)

void\* handle = dlopen (“libtest.so”, RTLD\_LAZY);

void (\*test)() = dlsym (handle, “my\_function”);

(\*test)();

dlclose (handle);

Para compilar necesitamos poner -ldl

g++ holaadios.cpp -o holaadios –ldl

**Processes**

**Looking at Processes**

Los procesos hijo heredan las tuberías padre.

Los paréntesis abren un subproceso: (Se duplica la terminal). Por ejemplo: **(cd pepe).**

**Ejecutar el proceso en segundo plano se utiliza al final el &**

**(echo “Hola” && sleep 3 && echo “adios”)&**

Un proceso que se ejecuta en segundo plano se llama **job**, que tienen un number

**Jobs** nos muestra un listado de todos los detenidos y ejecutándose.

Detener un proceso con ctrl + z.

Como levantar un proceso que está en segundo plano (fg y el número del proceso).

**Process IDs**

#include <stdio.h>

#include <unistd.h>

int main ()

{

printf (“The process ID is %d\n”, (int) getpid ());

printf (“The parent process ID is %d\n”, (int) getppid ());

return 0;

}

**Viewing Active Processes**