|  |  |  |
| --- | --- | --- |
| GCC GBD (Comandos) | La función que realiza | Ejemplo |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GCC  Comandos mas comunes $ gcc main\_es.c  $ ./a.out  $ gcc -o main\_es main\_es.c  GBD  Comandos mas comunes  ~$ ./lista  ~$ gdb lista  ~$ gcc -o lista lista1.c -std=c99 -Wall -DTEST --debug   |  | | --- | | .c | | .C .cc  .cpp  .c++  .cp  .cxx | | .m | | .i | | .ii | | .s | | .o | | .h |   GNU gdb (GDB) 7.4.1-debian  Copyright (C) 2012 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law. Type "show copying"  and "show warranty" for details.  This GDB was configured as "i486-linux-gnu".  For bug reporting instructions, please see: | Abre el fichero con el editor y comprueba que contiene la función main por la que comienza la ejecución del programa así como una función auxiliar. A continuación ejecuta el comando para transformar este fichero de código en un programa:   El compilador está diseñado para que si no se especifica el nombre del programa a crear, por defecto lo deja en a.out.  Tras ejecutar este comando comprueba que tienes **dos** ejecutables en el directorio actual. Borra el que ya no necesitas.    Para correr el programa simplemente se ejecuta.  Lista es el nombre de nuestro programa. Si el programa recibe argumentos, pueden pasarse aquí o pueden setearse luego.  Para compilar, se puede correr el siguiente comando, -DTEST define la macro TEST para que se incluya la función main, que permite testear la funcionalidad de la lista.   |  | | --- | | fuente en C | | fuente en C++; se recomienda .cpp | | fuente en Objective-C | | C preprocesado | | C++ preprocesdo | | fuente en lenguaje ensamblador | | código objeto | | archivo para preprocesador (encabezados), no suele figurar en la linea de comando de gcc | | gcc hola.c  compila el programa en C hola.c, gener un archivo ejecutable a.out.  gcc -o hola hola.c  compila el programa en C hola.c, gener un archivo ejecutable hola.  g++ -o hola hola.cpp  compila el programa en C++ hola.c, gener un archivo ejecutable hola.  gcc -c hola.c  no genera el ejecutable, sino el código objeto, en el archivo hola.o. Si no s indica un nombre para el archivo objeto, usa el nombre del archivo en C y le cambia la extensión por .o.  gcc -c -o objeto.o hola.c  genera el código objeto indicando el nombre de archivo.  g++ -c hola.cpp  igual para un programa en C++.  g++ -o ~/bin/hola hola.cpp  genera el ejecutable hola en el subdirectorio bin del directorio propio del usuario.  g++ -L/lib -L/usr/lib hola.cpp  indica dos directorios donde han de buscarse bibliotecas. La opción -L debe repetirse para cada directorio de búsqueda de bibliotecas.  g++ -I/usr/include hola.cpp  indica un directorio para buscar archivos de encabezado (de extensión .h).  debugger GDB  Compilar programas en C con opcion -g  % gcc -g prog.c ... -o <archivo binario.  Si no se especifica -o, se genera el archivo a.out  + Gdb se invoca con  gdb <archivo binario>  + Para obtener ayuda:  help <comando>  + Poner breakpoint en funciones:  (gdb) b main  + Borrar breakpoints:  (gdb) del <nro. del break>  + Mostrar nros. de breakpoint  (gdb) info break  + Correr el programa:  (gdb) run  + Ver el encadenamiento de funciones de la tarea actual  (la que le tiene la CPU en ese momento):  (gdb) where  + subir y bajar en la pila para ver variables de funciones intermedias:    (gdb) up -> Si F llamo a G y estamos en G, pasa a F  (gdb) down -> vuelve a G  + Especifico para programas en C:  - Ejecutar paso a paso instrucciones en C (step y next):  (gdb) s -> ejecuta una instruccion. Si hay una llamada a una funcion,  se detiene en la primera instruccion de esa funcion.  (gdb) n -> ejecuta una instruccion. Si hay llamadas a funciones, se  las ejecuta completamente sin detenerse.  - Imprimir valores de expresiones  (gdb) p x->a.d + 1 ( p de print )  - Imprimir las variables locales de la funcion examinada:  (gdb) info locals  + Especifico para programas en Assembler:  - Ejecutar paso a paso instrucciones de maquina:  (gdb) stepi -> ejecuta una instruccion de maquina.  (gdb) nexti -> si es un call, ejecuta el call hasta el retorno,  si no, ejecuta una instruccion de maquina.  Ambas instrucciones muestran la direccion de la siguiente instruccion  a ejecutar en hexadecimal. Ej.:  (gdb) 0x080483d4 in dump ()  - Para mostrar 10 instrucciones de maquina a partir de una direccion |