



Sistemas de Computación (Ing. en Computación)

Guía rápida para uso del debugger GDB

1. Compilar programas con opción -g

```
% gcc -g prog.c ... -o <archivo binario>
```

Si no especifica -o, se genera el archivo a.out

2. Invocar gdb con el siguiente comando:

```
gdb <archivo binario>
```

ó

```
gdb -q <archivo binario> ; No muestra banner de gdb
```

3. Principales comandos disponibles:

```
help <comando> ; Para obtener ayuda:
```

```
break main ; breakpoint en función mail
```

```
b main
```

```
b #linea ; breakpoint en #linea
```

```
delete <nro. del break> ; Borra breakpoint
```

```
info break ; Mostrar nros. de breakpoints
```

```
run ; Corre el programa
```

```
where ; Ver encadenamiento de funciones de  
la tarea actual (la que tiene la  
CPU en ese momento)
```

```
up ; Subir en la pila,  
Si F call G y estamos en G, pasa F
```

```
down ; Bajar en la pila, vuelve a G
```

```
info register ; Muestra contenido de registros
```

4. Específico para programas en C:

```
list ; Lista programa fuente
```

```
disass main ; Desensambla función main
```

```
set variable=algo ; Cambiar valor de variable
```

```
step ; Si hay una llamada a una función,  
entra a la función y se detiene en  
la primera instrucción.
```

```
shell ; Ejecuta comandos del SO
```

```
start ; Ejecuta línea por línea
```

```
next ; Ejecuta una instrucción. Si hay  
Llamada a función la ejecuta sin  
Detenerse.
```

```
continue ; Continúa con la ejecución hasta fin
```

```
print ; Imprimir valores de expresiones,
```



```

p                                ; variables y funciones.
                                ; Depende contexto.
                                ; Eje.: p x->a.d + 1
                                ; print sEdad
                                ; print FnMensaje("Juan", 89)

ptype variable                  ; devuelve el tipo de la variable
Ptype función                  ; devuelve firma de la función

info locals                     ; Muestra variables locales y su
                                contenido en el contexto.
display variable                ; Muestra valor variable después de
                                de cada next

i r sp                          ; Lista el contenido de SP

GDB proporciona un pseudo-registro $sp que se asigna automáticamente
al registro de puntero de pila correcto para una plataforma
determinada.

x/32xw $sp                      ; Lista 32 posiciones de memoria a
                                partir del valor de sp

```

5. Específico para programas en Assembler:

- Ejecutar paso a paso instrucciones de máquina:

```

stepi                          ; Ejecuta una instrucción de máquina.
Nexti                          ; Si es un call, lo ejecuta hasta el
                                retorno, si no, ejecuta solo una
                                instrucción de máquina.

```

Ambas instrucciones muestran la dirección de la siguiente instrucción a ejecutar en hexadecimal. Ej.:

```
0x080483d4 in dump ()
```

```

x/10i 0x080483d2                ; Muestra 10 instrucciones de máquina
                                ; a partir de la dirección indicada

```

```

0x80483d2 <dump+6>:      mov    %ebp,%eax
0x80483d4 <dump+8>:      mov    %eax,0xffffffff(%ebp)
0x80483d7 <dump+11>:     mov    0xffffffff(%ebp),%eax

```

```

x/10x 0xbffff5c8            ; Muestra contenido a partir de una
                                dirección hexadecimal

```