

CSE306 Software Quality in Practice

Dr. Carl Alphonse
alphonse@buffalo.edu
343 Davis Hall

More gdb commands

- C-x C-a toggle between a "graphical" and line-based UI
- `break <line>` (e.g. `break 31`)
- `info b` (list breakpoints)
- `c` (continue to next breakpoint), `c 10`
- `watch <variable>` (e.g. `watch i`)
 - <https://sourceware.org/gdb/current/onlinedocs/gdb/Set-Watchpoints.html#Set-Watchpoints>
- Looking at source code:
 - `list line#`
 - `list function`
 - `disassemble /m`
- Looking at data:
 - `print`
 - `examine (x)`
 - `x /s name`, `x/48c name` (addresses in hex!)
- <https://sourceware.org/gdb/current/onlinedocs/gdb/Memory.html#Memory>

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

int main(int argc, char * argv[]) {

    if (argc !=2) {
        printf("Please give one numeric argument.\n");
        return 1;
    }

    int limit = atoi(argv[1]);

    char * string,* name;
    name = malloc(3 * sizeof(*name));
    string = malloc(9 * sizeof(*string));
    name[0] = '@';
    name[1] = '$';
    name[2] = '\0';
    string[0] = 's';
    string[1] = 'e';
    string[2] = 'r';
    string[3] = 'e';
    string[4] = 'n';
    string[5] = 'i';
    string[6] = 't';
    string[7] = 'y';
    string[8] = '\0';
    printf("string has length %zu and is %s.\n",strlen(string),string);
    printf("name has length %zu and is %s.\n",strlen(name),name);
    for (int i=3; i<limit; i++) {
        name[i] = (char) ('a'+((i-3)%26));
    }
    name[limit] = '\0';
    printf("string has length %zu and is %s.\n",strlen(string),string);
    printf("name has length %zu and is %s.\n",strlen(name),name);
    return 0;
}
```

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

```
int main(int argc, char * argv[]) {
```

```
    if (argc !=2) {
        printf("Please give one numeric argument.\n");
        return 1;
    }
```

```
    int limit = atoi(argv[1]);
```

```
    char * string,* name;
    name = malloc(3 * sizeof(*name));
    string = malloc(9 * sizeof(*string));
    name[0] = '@';
    name[1] = '$';
    name[2] = '\0';
    string[0] = 's';
    string[1] = 'e';
    string[2] = 'r';
    string[3] = 'e';
    string[4] = 'n';
    string[5] = 'i';
    string[6] = 't';
    string[7] = 'y';
    string[8] = '\0';
```

```
    printf("string has length %zu and is %s.\n",strlen(string),string);
```

```
    printf("name has length %zu and is %s.\n",strlen(name),name);
```

```
    for (int i=3; i<limit; i++) {
        name[i] = (char) ('a'+((i-3)%26));
    }
```

```
    name[limit] = '\0';
```

```
    printf("string has length %zu and is %s.\n",strlen(string),string);
```

```
    printf("name has length %zu and is %s.\n",strlen(name),name);
```

```
    return 0;
```

```
}
```

GitHub Classroom link for this code:
<https://classroom.github.com/a/UZDp1zg->
(also posted on schedule page of course website)

```
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> gcc -std=c11 -Wall -g -o prg exercise-memory.c
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> prg 4
prg: Command not found.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 4
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 8 and is serenity.
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 20
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 8 and is serenity.
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 30
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 8 and is serenity.
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 31
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 8 and is serenity.
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 32
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 0 and is .
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> ./prg 40
string has length 8 and is serenity.
name has length 2 and is @$.
string has length 8 and is defghijk.
name has length 2 and is @$.
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc>
```

```
cerf:~/CSE306/code/intermediate-gdb-carl-alphonc> gdb prg
(gdb) run 33
Starting program: /home/csefaculty/alphonc/CSE306/code/intermediate-gdb-carl-alphonc/prg 33
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
string has length 8 and is serenity.
name has length 2 and is @$.
```

```
Breakpoint 1, main (argc=2, argv=0x7fffffff988) at exercise-memory.c:31
```

```
31 for (int i=3; i<limit; i++) {
```

```
(gdb) info b
```

Num	Type	Disp	Enb	Address	What
1	breakpoint	keep y		0x0000555555555303	in main at exercise-memory.c:31

breakpoint already hit 1 time

```
(gdb) info frame
```

```
Stack level 0, frame at 0x7fffffff880:
```

```
rip = 0x555555555303 in main (exercise-memory.c:31); saved rip = 0x7ffff7da9d90
source language c.
```

```
Arglist at 0x7fffffff870, args: argc=2, argv=0x7fffffff988
```

```
Locals at 0x7fffffff870, Previous frame's sp is 0x7fffffff880
```

```
Saved registers:
```

```
rbp at 0x7fffffff870, rip at 0x7fffffff878
```

```
(gdb) into locals
```

```
Undefined command: "into". Try "help".
```

```
(gdb) info locals
```

```
i = 0
```

```
limit = 33
```

```
string = 0x5555555592c0 "serenity"
```

```
name = 0x5555555592a0 "$@"
```

```
(gdb)
```