CS 136L Lecture 12 Module 9

GDB

Setting Up GDB

- `layout src`
- `winheight cmd +5` or `winheight src +5`
- `set style enable off` (disables colouring)
- `set logging file [filename]` then `set logging on` or `off`

Program

- `run`
- `break main` (or line number)
- `info breakpoints`
- `delete 1`
- `enable ID` or `disable ID`
- `next` or `n`
- `refresh` (if screen goes wonky)
- `print` (variable name) `whatis` (prints variable type)
- `watch` and `info watchpoints` (stops when expr changes values)
- `set var is_complete=False` sets the variable `is_complete` to be `False`

Demo 1 (9.3 on EdX)

```
int baz(int *p){return *p + 1;}
int bar(int *p, int val){
  int *x = NULL;
  if ( val % 2 ){ x = &val;}
  else {x = p;}
  return baz(x);
}
int foo(int *p){
   int ans = 0;
   int *q = NULL;
   if (*p < 5 ){
     ans = bar(p, 2 * *p);
   } else {
     p = q;
     int temp = 42;
     ans = bar(p, temp);
   } return ans;}
```

Demo 1 (Continued)

```
int main(void){
    int x = 10;
    x = foo(&x);
    if (x == 0) {
      printf("Error: x set to 0\n");
      return 3;
    } else {
      printf("Result %d\n",x);
    }
}
```

Sample

```
break main
run
continue
backtrace  #reach error
up
print x
up
print p
up
break foo
delete 1
run
display p
n #5 times
```

Sample

```
break main
run
record
continue
reverse-next
reverse-next
list #(if not using src)
print x
reverse-next # 6 times
print p
list
reverse-next
print p
print q
```

Demo 2

```
int factorial(const int n){
  if(n){
    return n * factorial(n - 1);
  }else{ // Zero Base Case
    return 1;
  }}
int main(){
  int n;
  printf("Please enter a positive integer: "
   );
  if (scanf("%d", &n) == 1 && n >= 0) {
    printf("%d! = %d\n", n, factorial(n));
  }else{
    printf("That's not a positive integer!\n
   ");
  }}
```

Demo 2

```
break factorial
run
continue # times 3
#maybe set var n=3 again to see.
backtrace
finish  # 4 times
```