Computer Systems Organization

PROF. MOHAMED ZAHHRAN
TA SESSION 2 - GNU DEBUGGER
PRESENTED BY: MAHLA RAHIMINEJAD
Virtual machine
Unix tutorial
Gdb
objcumb
Virtual machine
Virtual machine (cont’d)

CSO Fall 2014
Welcome to Ubuntu!
Unix commands

First step: open terminal
Unix useful commands (cont’d)

- **pwd:** show in which directory you located
- **ls:** show the list of files in current directory
- **cd:** allow us to change directory
- **cd .. :** navigate up one level
- **mkdir:** create directories
- **./filename.extension:** execute a file
How to compile c file?

Step 1: go to the code directory

Step 2: gcc -g code_name.c

Step 3: enjoy the errors!
Oops!

Time to debug!

Segmentation fault

Unidentified functions!

Pointers...

Unexpected output!
gdb!

Gnu debugger

Let you see what’s going on in your program!

It can
- Start running your program
- Make your program stop on specified condition
- Examine what has happened
- Change things!
Gdb (cont’d)

cdorg@orgmachine:~$ cd Downloads/
org@orgmachine:~/Downloads$ gcc -g samplecode.c -o samplecode
org@orgmachine:~/Downloads$ gdb samplecode
GNU gdb (GDB) 7.2-ubuntu
Copyright (C) 2010 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 "i686-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/org/Downloads/samplecode...done.
(gdb)
Gdb (cont’d)

```
Breakpoint 1, main () at samplecode.c:20
20      a = 1;
(gdb) s
22      printf("welcome to my program!\n");
(gdb) s
welcome to my program!
23      while (count < 12){
(gdb) s
24          count_func (&count);
(gdb) s
(count_func (cnt=0xbfff7ac) at samplecode.c:9
9          *cnt = *cnt + 1;
(gdb) s
11     }
(gdb) s
main () at samplecode.c:25
25      printf("%d ", count );
(gdb) n
26      printf("%d ", a );
(gdb) p a
$1 = 1
(gdb)
```
gdb useful commands

- `gdb <exe file>`: start debugger
- `r`: run
- `b #line num`: breakpoint
- `b <function name>`: breakpoint at start of function `<function_name>`
- `b #line #cond`: conditional breakpoint
- `disable`: disable breakpoint
- `disp <variable name>`: display amount of variable
- `undisplay`: display no more!
gdb useful commands (cont’d)

- **p <variable name>**: print variable value
- **n/nexti**: go to next line
- **s/stepi**: step through function, loop, etc
- **Call func (a , b , ...)**: call func with a , b, ... and print return value
- **disas**: disassemble current function
- **x/[num][size][format]**: examine [num] number of objects, each of them with size [size] and display them in format [format]
References


http://csapp.cs.cmu.edu/public/docs/gdbnotes-ia32.pdf

http://www.ee.surrey.ac.uk/Teaching/Unix/

http://www.gnu.org/software/gdb/

https://sourceware.org/gdb/download/onlinedocs/
Thank you! ;}