Integrazione e testing di ChatGPT in LLDB debugger: verso un debugging automatico

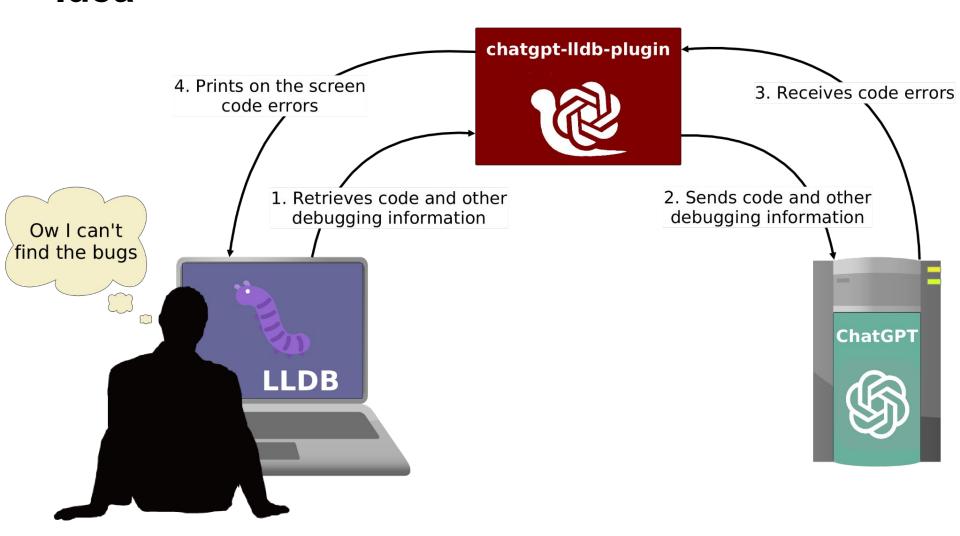
Facoltà di Ingegneria dell'Informazione Informatica e Statistica Laurea in Ingegneria dell'Informazione



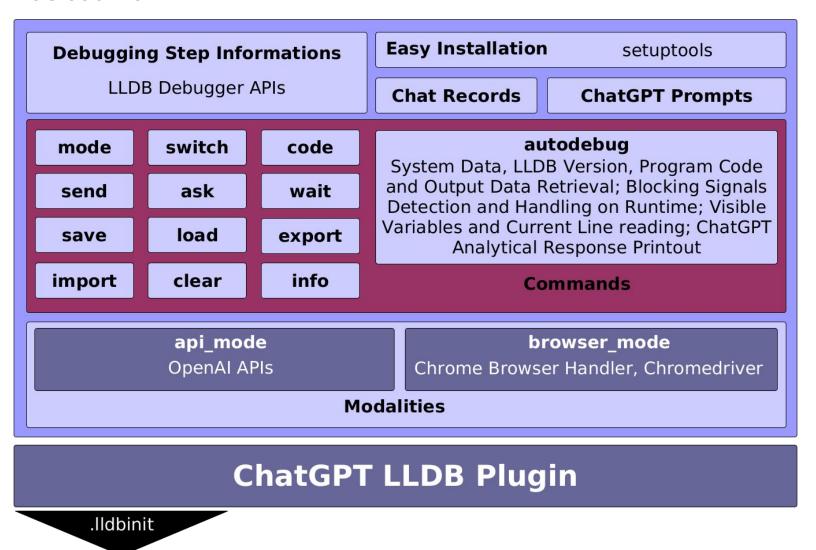
Carlo Capodilupo Matricola 1851916 Relatore
Prof. Giuseppe Antonio <u>Di Luna</u>

Problema

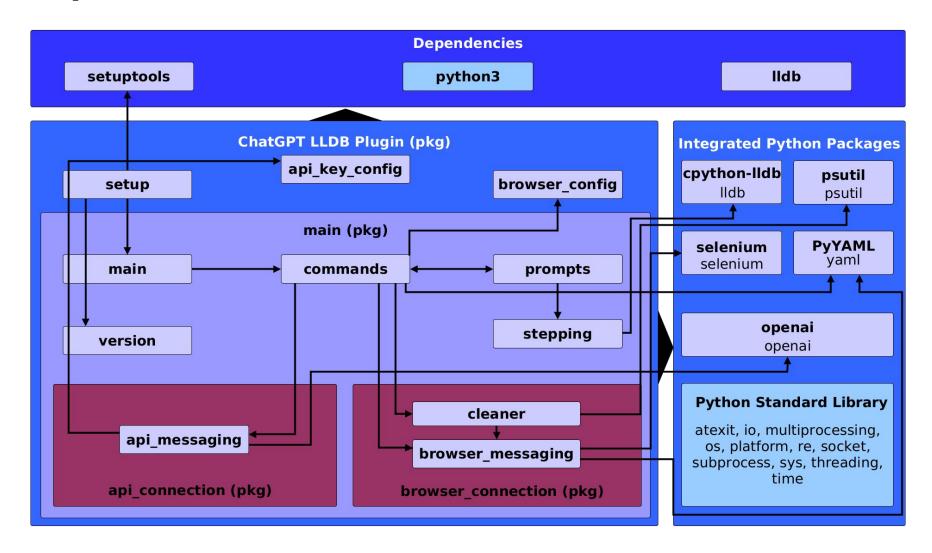
Idea



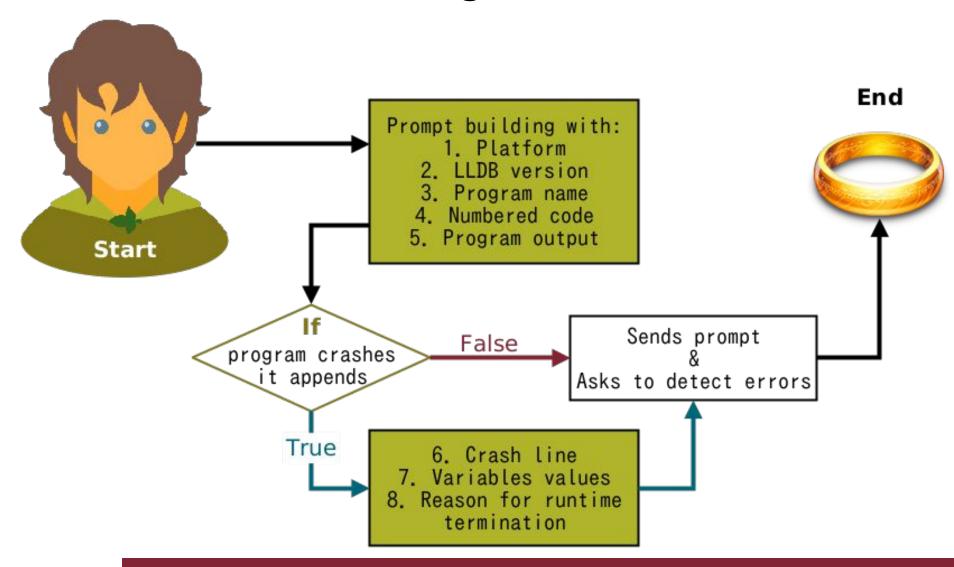
Architettura



Dipendenze



Il comando "autodebug"



Codice

github.com/lupidux/chatgpt-lldb-plugin



```
mig detected
def automated debugging (debugger):
      target = debugger, GetSetectedTarget()
      target.DeleteAllEreakpoints[]
     target.SreakpointCreateSyName("main")
     process = target.GetProcess()
if process.IsValid():
           process Kill()
     executable dir = target DetExecutable() G
     executable filenees - target.GetExecutable executable path - cs.path.join(executable
      proc - subprocess. Popen(executable path
     cutput, error = proc.communicate()
cutput string = cutput = error
cutput string = cutput string.replace
      process - target.LaunchSimple(None,
              process.GetState() to Uldb.aSt.
           debagging file = get sourcefile ;
line of crush = get current line(
thread = process.SetThreadAtIndes
            thread.Step@ver[]
            stop reason - thread.GetStopReason()
                stap reason - 11db eSta
                stop / restor — Ltds.e5copResearcip,
signal info = throat.LestStopReseri,
st "SIGABHT" in signal_info:
    sig detected = "SIGABHT"
    signal_info:
    sig_detected = "SIGABGY indirizes non
    sig_detected = "SIGSEGY indirizes non
                       frace = target.GetProcess().GetSelectedTh
                       address = frome.GetPCAddress[]
                       memory data = target.ReadMemory|addres
11 memory data in Norms:
| mamory_read_failed = True
                             variables to be sent
                       rig detected = "SIGFPE"
                        sig detected = sign
                       terget_DeleteA!

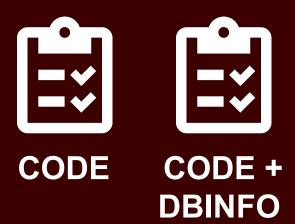
    Lidb.eStateStopped:

                                                   aStateExited and process.GetExitStatus() - 0:
```

I Sperimentazione:



II Sperimentazione:



Integrazione e testing di ChatGPT in LLDB debugger: verso un debugging automatico

26/10/2023

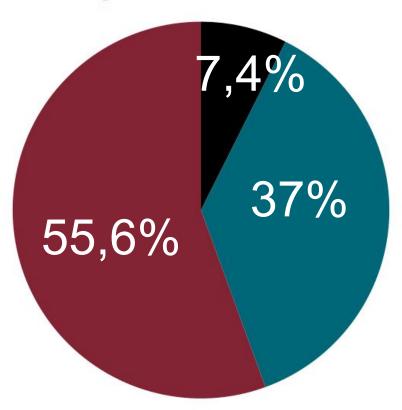
Pagina 8

Analisi dei risultati

I Sperimentazione

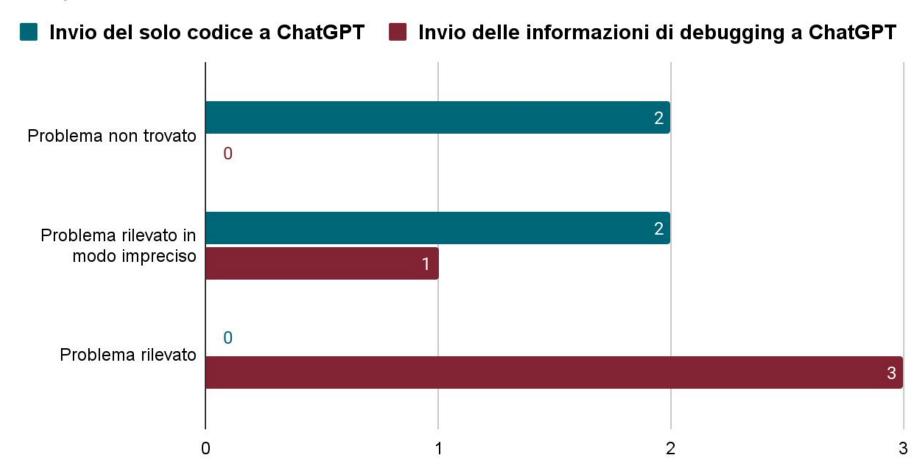
Problema non trovato
 Problema rilevato in modo impreciso

Problema rilevato



Analisi dei risultati

II Sperimentazione



I Sperimentazione

```
1 #include <stdio.h>
3 int main() {
      int *ptr = NULL;
      *ptr = 42;
      return 0;
```

I Sperimentazione



user@computer:~\$ Ildb esempio_1_CODE+DBINFO
user@computer:~\$ (Ildb) autodebug

Risposta:

Il segmentation fault (SIGSEGV) è causato dal tentativo di scrivere in un'area di memoria alla quale non si ha accesso. L'errore è causato dal puntatore nullo "ptr" che non punta a nessuna area di memoria valida.

II Sperimentazione

```
#include <stdio.h>
  #include <math.h>
3
  int main() {
      int a = 1;
5
      int b = 0;
      int c = log(pow(sqrt(a),b) - sin(b));
      int result = a / c;
      printf("Result: %d\n", result);
10
11
      return 0;
12
13
```

II Sperimentazione



```
user@computer:~$ Ildb esempio_2_CODE
user@computer:~$ (Ildb) autodebug
```



È necessario eseguire dei cast espliciti in quanto alcune delle funzioni matematiche utilizzate restituiscono dei valori in virgola mobile che sono poi assegnati a delle variabili di tipo "int".

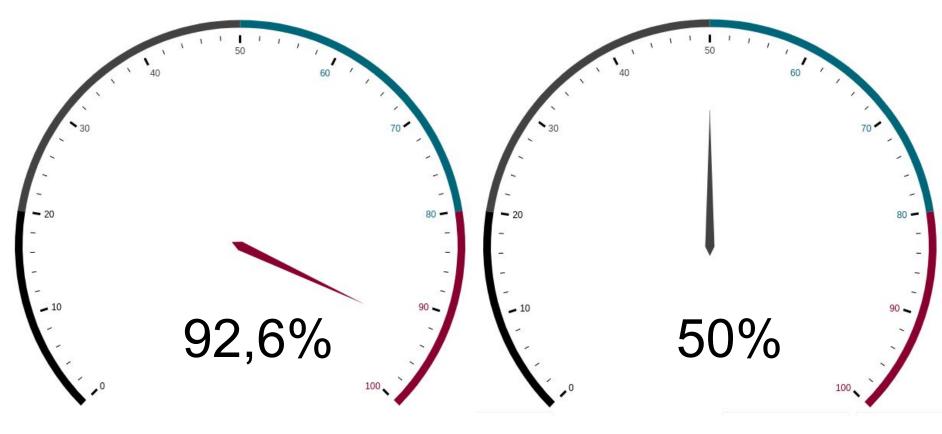


```
user@computer:~$ Ildb esempio_2_CODE+DBINFO user@computer:~$ (lldb) autodebug
```



Nella linea 9 del codice c'è un problema di divisione per zero. Il problema si verifica quando cerchi di dividere "a" per "c" in quanto "c" è 0.

Conclusioni



Bug rilevati

Vantaggio db info

Grazie per l'attenzione