Lab 4. Database and GDB

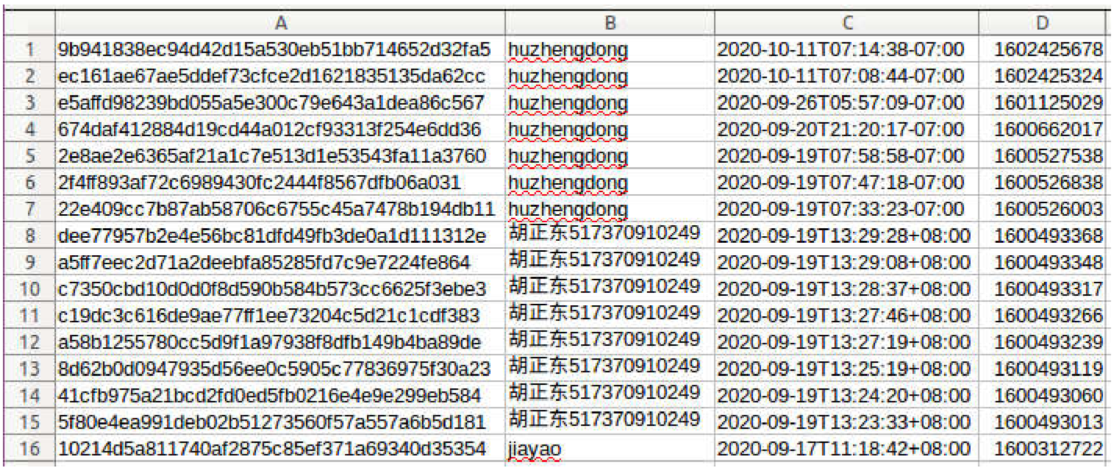
Hu Zhengdong 517370910249

Oct. 14, 2020

I. Database

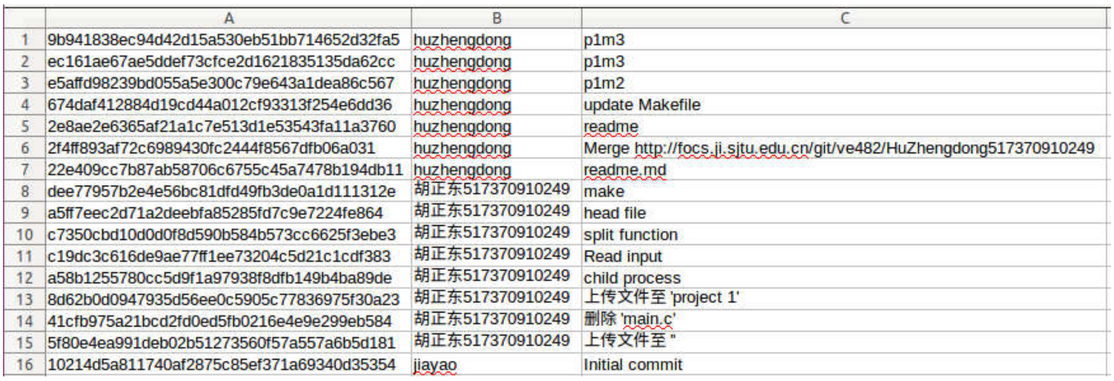
1. Database Creation

```
mumsh $ git log --pretty="%H,%aN,%aI, %at" > timestamp.csv
```



	A	B	C	D
1	9b941838ec94d42d15a530eb51bb714652d32fa5	huzhengdong	2020-10-11T07:14:38-07:00	1602425678
2	ec161ae67ae5ddef73cfce2d1621835135da62cc	huzhengdong	2020-10-11T07:08:44-07:00	1602425324
3	e5affd98239bd055a5e300c79e643a1dea86c567	huzhengdong	2020-09-26T05:57:09-07:00	1601125029
4	674daf412884d19cd44a012cf93313f254e6dd36	huzhengdong	2020-09-20T21:20:17-07:00	1600662017
5	2e8ae2e6365af21a1c7e513d1e53543fa11a3760	huzhengdong	2020-09-19T07:58:58-07:00	1600527538
6	2f4ff893af72c6989430fc2444f8567dfb06a031	huzhengdong	2020-09-19T07:47:18-07:00	1600526838
7	22e409cc7b87ab58706c6755c45a7478b194db11	huzhengdong	2020-09-19T07:33:23-07:00	1600526003
8	dee77957b2e4e56bc81dfd49fb3de0a1d111312e	胡正东517370910249	2020-09-19T13:29:28+08:00	1600493368
9	a5ff7eec2d71a2deebfa85285fd7c9e7224fe864	胡正东517370910249	2020-09-19T13:29:08+08:00	1600493348
10	c7350cbd10d0d0f8d590b584b573cc6625f3ebe3	胡正东517370910249	2020-09-19T13:28:37+08:00	1600493317
11	c19dc3c616de9ae77ff1ee73204c5d21c1cdf383	胡正东517370910249	2020-09-19T13:27:46+08:00	1600493266
12	a58b1255780cc5d9f1a97938f8dfb149b4ba89de	胡正东517370910249	2020-09-19T13:27:19+08:00	1600493239
13	8d62b0d0947935d56ee0c5905c77836975f30a23	胡正东517370910249	2020-09-19T13:25:19+08:00	1600493119
14	41cfb975a21bcd2fd0ed5fb0216e4e9e299eb584	胡正东517370910249	2020-09-19T13:24:20+08:00	1600493060
15	5f80e4ea991deb02b51273560f57a557a6b5d181	胡正东517370910249	2020-09-19T13:23:33+08:00	1600493013
16	10214d5a811740af2875c85ef371a69340d35354	jiayao	2020-09-17T11:18:42+08:00	1600312722

```
mumsh $ git log --pretty="%H,%aN,%s" > db.csv
```



	A	B	C
1	9b941838ec94d42d15a530eb51bb714652d32fa5	huzhengdong	p1m3
2	ec161ae67ae5ddef73cfce2d1621835135da62cc	huzhengdong	p1m3
3	e5affd98239bd055a5e300c79e643a1dea86c567	huzhengdong	p1m2
4	674daf412884d19cd44a012cf93313f254e6dd36	huzhengdong	update Makefile
5	2e8ae2e6365af21a1c7e513d1e53543fa11a3760	huzhengdong	readme
6	2f4ff893af72c6989430fc2444f8567dfb06a031	huzhengdong	Merge http://focs.ji.sjtu.edu.cn/git/ve482/HuZhengdong517370910249
7	22e409cc7b87ab58706c6755c45a7478b194db11	huzhengdong	readme.md
8	dee77957b2e4e56bc81dfd49fb3de0a1d111312e	胡正东517370910249	make
9	a5ff7eec2d71a2deebfa85285fd7c9e7224fe864	胡正东517370910249	head file
10	c7350cbd10d0d0f8d590b584b573cc6625f3ebe3	胡正东517370910249	split function
11	c19dc3c616de9ae77ff1ee73204c5d21c1cdf383	胡正东517370910249	Read input
12	a58b1255780cc5d9f1a97938f8dfb149b4ba89de	胡正东517370910249	child process
13	8d62b0d0947935d56ee0c5905c77836975f30a23	胡正东517370910249	上传文件至 'project 1'
14	41cfb975a21bcd2fd0ed5fb0216e4e9e299eb584	胡正东517370910249	删除 'main.c'
15	5f80e4ea991deb02b51273560f57a557a6b5d181	胡正东517370910249	上传文件至 ''
16	10214d5a811740af2875c85ef371a69340d35354	jiayao	Initial commit

2. Database system installation

- What are the most common database systems?
  Oracle, Mysql, SQL Server, MongoDB, PostgreSQL

- Briefly list the pros and cons of the three most common ones.
  - Oracle
    - Pros: Oracle database management are incredibly robust
    - Cons: The cost of Oracle can be prohibitive and require significant resources
  - Mysql
    - Pros: Mysql is free for use, and there are a variety of user interfaces
    - Cons: Mysql is lack of some automatic functions, and there is no built-in support for XML or OLAP
  - SQL Server
    - Pros: SQL server is fast and stable, and offers the ability to adjust and track performance levels
    - Cons: Enterprise pricing is beyond what many organizations can afford, and many individuals have issues using the SQL server integration
- Create an empty SQLite database

```
sqlite3 test.db
```

- Use the SQLite shell to prepare two empty tables for each of your .csv file
- Import each .csv file in its corresponding SQLite table. (actually .psv file)

```
//Create the table db
CREATE TABLE db(
HashCommit TEXT PRIMARY KEY,
AuthorName TEXT,
Subject TEXT);

//Import the db.psv
.separator ‘|’
.import db.psv db

//Create the table timestamp
CREATE TABLE timestamp(
HashCommit TEXT PRIMARY KEY,
AuthorName TEXT,
ISODate TEXT,
UnixDate INT);

//Import the db.psv
.separator ‘|’
.import timestamp.psv timestamp
```

3. Database queries

- Who are the top five contributors to the Linux kernel since the beginning?

```
SELECT AuthorName, count(*) AS count
FROM db
GROUP BY AuthorName
ORDER BY count DESC
LIMIT 5;
```

Top 5:
Linus Torvalds, David S. Mille, Takashi Iwai, Mark Brown, Arnd Bergmann

- Who are the top five contributors to the Linux kernel for each year over the past five years?

```
SELECT AuthorName, count(*) AS count, strftime('%Y', ISODate) as Year  FROM timestamp
WHERE Year== “2020”
GROUP BY AuthorName
ORDER BY count DESC
LIMIT 5;
```

Change the value of Year to get the top 5 contributors for each year over the past five years

- What is the most common “commit subject”?

```
SELECT Subject, count(*) AS count FROM db
GROUP BY Subject
ORDER BY count DESC
LIMIT 1;
```

The most common:
"Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net"

- On which day is the number of commits the highest?

```
SELECT count(*) AS count, strftime('%Y-%m-%d', ISODate) as Date  FROM timestamp
GROUP BY Date
ORDER BY count DESC
LIMIT 1;
```

The day of the highest number of commits
2008-01-30

- Determine the average time between two commits for the five main contributor?

```
SELECT AuthorName, count(*) AS count, (max(UNIXDate)-min(UNIXDate))/(count(*)-1) FROM Time
GROUP BY AuthorName
ORDER BY count DESC
```

```
LIMIT 5;
```

II. Debugging

- How to enable built-in debugging in gcc?

```
gcc -g
```

- What is the meaning of GDB?

GNU Debugger

- Compile the master branch of you mumsh with debugging enabled.

Add -g to the compile instructions in Makefile

1. Basic GDB usage

- Find the homepage of the GDB project.

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

- What languages are supported by GDB?

Ada, Assembly, C, C++, D, Fortran, Go, Objective-C, OpenCL, Modula-2, Pascal, Rust

- What are the following GDB commands doing?

• backtrace

To print a backtrace of the entire stack

• where

print out information about what's the current call frame is

• finish

continue execution of the current function until it returns to its caller.

• delete

Deletes specified breakpoints or all breakpoints.

• info breakpoints

Print the information about the breakpoints

- Search the documentation and explain how to use conditional breakpoints

gdb allows the definition of conditional breakpoints. These points are defined by first creating a regular breakpoint, and then attaching a condition. The execution is suspended if the breakpoint is reached and the condition satisfied. If the condition is not satisfied, the breakpoint has no effect.

Example:

```
(gdb) b 29    //Set a breakpoint in the 29th line which is the 4th breakpoint
```

```
(gdb) condition 4 y==9999 //set the condition for the 4th breakpoints
```

- What is -tui option for GDB?
  Activate the Text User Interface when starting. The Text User Interface manages several text windows on the terminal, showing source, assembly, registers and gdb command outputs

- What is the "reverse step" in GDB and how to enable it. Provide the key steps and commands.

"reverse step" in GDB is to reverse debugging in GDB.

The key step is to enter "record" to enable "reverse-step", then gdb continues debugging which can be reversed afterwards.

Key commands:

- reverse-continue ('rc') -- Continue program being debugged but run it in reverse
- reverse-finish -- Execute backward until just before the selected stack frame is called
- reverse-next ('rn') -- Step program backward, proceeding through subroutine calls.
- reverse-nexti ('rni') -- Step backward one instruction, but proceed through called subroutines.
- reverse-step ('rs') -- Step program backward until it reaches the beginning of a previous source line
- reverse-stepi -- Step backward exactly one instruction
- set exec-direction (forward/reverse) -- Set direction of execution.