

# C Programming II

## 2024 Spring

### Homework 04

Instructor: Po-Wen Chi

Due: 2024.06.02 PM 11:59

#### Policies:

- **Zero tolerance** for late submission.
- **Plagiarism is not allowed.** Both source and copycat will be **zero**.
- You need to prepare a README file about how to make and run your program. Moreover, you need to provide your name and your student ID in the README file.
  - Your Name and Your ID.
  - The functional description for each code.
  - Anything special.
- Please pack all your submissions in one zip file.
- For convenience, your executable programs must be named following the rule hw**XXYY**, where the red part is the homework number and the blue part is the problem number. For example, hw0102 is the executable program for homework #1 problem 2.
- I only accept **PDF**. MS Word is not allowed.
- **Do not forget your Makefile.** For convenience, each assignment needs only one Makefile.

## 1 System Monitor(30 pts)

**top** command is used to show the Linux processes. It provides a dynamic real-time view of the running system. You can try this command on your computer. This time, I want you to develop a program similar to **top**. Do not worry, I will show you how to do this.

The proc filesystem (**procfs**) is a special filesystem in Unix-like operating systems that presents information about processes and other system information. It provides a more convenient and standardized method for accessing process data **held in the kernel** than traditional tracing methods or direct access to kernel memory. What you should do is to

find all numeric directories in `/proc`. Each process in your computer will have a directory here and you can get the process information in the directory. For your simplicity, I only want you to read `/proc/[pid]/stat`. You can read the proc manual<sup>1</sup> to know how to parse this file.

```
1 $ ./hw0401 --help
2 Usage: hw0401 [options]
3   -t, --time-interval=time    Update the information every [time] seconds.
   Default: 5 seconds.
4   -c, --count                Update the information [count] times. Default:
   infinite.
5   -p, --pid=pid              Only display the given process information.
6   -h, --help                 Display this information and exit.
7 $ ./hw0401
8 PID   NAME      state  CPU    MEM
9 1    systemd    S      1.00%  174088192
10 2    kthreadd   S      2.00%   0
11 3    rcu_gp     I      1.00%   0
12 ...
```

For your simplicity, I give you some hints:

- Memory usage for a process is its **vsiz**e.
- CPU usage calculation: <https://jihoony.gitbook.io/developers-notes/developer/c-c++/calculate-cpu-usage-on-linux-as-top>
- man **opendir**.

## 2 How to Rent your Bike? (30 pts)

Have you ever ride **Youbike** across Taipei City? Do you ever feel unsettling when you haven't check the status of the Youbike station when you tend to rent or return your bike? Don't worry! Now you are going to create an App to do this for you!

In this session, you are going to develop a C program to check the status of Youbike stations across Taipei City. The program should be able to search the station by uid, the English name of the station, or any other features you think is necessary. The program should show the status of the stations, including the number of available bikes, the number of available parking slots, and the last update time of the data source.

### 2.1 Requirements

- Every interaction should be done in command line, and only C program is allowed.
- The executable file should be named hw0402 after compilation (make).
- You should provide hw0402\_README to list every feature you have implemented. Only .txt, .md, .pdf are acceptable.

---

<sup>1</sup>man proc

- Every external library you use should put license information in hw0402\_README as well. If caught without providing the license information, you will get a 0 for this homework.
- Your program should follow good coding practices, including proper indentation, use of comments, and meaningful variable names.

## 2.2 Basic Features (25 pts)

- Provide a **-h** function to show how to use the program (+5)
  - This should include all the available parameters and their descriptions.
- The program should be able to search the station by uid (+5)
- The program should be able to search the station by the English name of the station (+5)
  - Include the snaen and aren fields to search the station.
  - Implement substring search
  - The maximum number of stations to show is 5
- The program should show the status of every station (+5)
  - Station Name and/or uid
  - Number of available bikes
  - Number of available parking slots
  - Last update time of the data source
- Print error message upon invalid input (+5)
  - Examples: "No such station found", "Invalid input", etc.

## 2.3 Additional Features (up to 15 pts)

- Show Station Name in Chinese (+5)
- Support Youbike stop status search in Chinese (+5)
- Support fuzzy search for station name (+5)
  - Consider both uppercase and lowercase inputs.
  - e.g. "Gongguan Campus" can be searched by "GongGuan" or "campus".
- Support Youbike stop status search by latitude/longitude to show the 5 nearest stops (+5)
- Support Youbike stop status search from New Taipei City as well (+5)

- UI/UX tidiness (+0 5, graded by TA)
  - e.g., color, font, layout, parallelism, etc.

## 2.4 API Usage

How to get the data? We can use Application Programming Interface (API) to get the status from official sources. The API is provided by the Taiwan government, with colaboration with the Taipei City government, and the data is open to the public. The API is free to use, and you can access the data by sending an request to the API endpoint.

The API provides the real-time status of Youbike stations across Taiwan. One of the API endpoints is as follows:

[https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike\\_immediate.json](https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/youbike_immediate.json)

The data structure example is as follows, and you may need to find the documentation or fourm yourself to double-check:

```

1 [
2   ...
3   {
4     "sno": "500105061",
5     "sna": "YouBike2.0_師範大學公館校區_1",
6     "sarea": "文山區",
7     "mday": "2024-05-10 11:16:19",
8     "ar": "師大公館校區校門口(汀州路側)",
9     "sareaen": "Wenshan Dist",
10    "snaen": "YouBike2.0_National Taiwan Normal University Gongguan
11    Campus_1",
12    "aren": "National Taiwan Normal University Gongguan Campus (Tingzhou
13    Rd.)",
14    "act": "1",
15    "srcUpdateTime": "2024-05-10 11:16:24",
16    "updateTime": "2024-05-10 11:16:52",
17    "infoTime": "2024-05-10 11:16:19",
18    "infoDate": "2024-05-10",
19    "total": 34,
20    "available_rent_bikes": 14,
21    "latitude": 25.00753,
22    "longitude": 121.53719,
23    "available_return_bikes": 20
24  },
25  ...
26 ]

```

Some fields are clarified as follows:

- **mday**: Update time of the data source for each station.
- **srcUpdateTime**: Time when the system publishes data updates.
- **updateTime**: Time when the data is stored in the database after processing by the platform.

- **infoDate, infoTime:** Update time of the data source for each station.

## 2.5 Sample Interaction

The examples below are sample interactions of the program.

```

1 $ ./hw0402 -h
2 Usage: ./hw0402 [OPTIONS]
3 Options:
4   -h                      Show this help message
5   -N <station_name>       Search station by name
6   -U <uid>                 Search station by UID
7   --sna <station_name>    Search station by name in Chinese
8   --lat <latitude> --lon <longitude> Search nearest stations by latitude and
    longitude

```

You can design your own interaction as long as it meets the requirements. Here I provide two different ways to interact with the program. Both are acceptable, but you may choose one to implement.

The first one:

```

1 $ ./hw0402
2 welcome to Taipei Youbike Station Status Checker!
3 search by station (N)ame, (U)id, (L)atitute and longitude: N
4 please enter the station name: Gongguan Campus
5 (show at most 5 stations with the substring "Gongguan Campus")
6
7 3 station(s) found with the substring "Gongguan Campus" in snaen and aren:
8
9 UID: 500105015
10 Station Name: YouBike2.0_National Taiwan Normal University Gongguan Campus
11 Available Bikes: 14
12 Available Parking Slots: 20
13 Last Update Time: 2024-05-10 11:16:24
14
15 UID: 500105061
16 Station Name: National Taiwan Normal University Gongguan Campus (Tingzhou Rd.)
17 Available Bikes: 12
18 Available Parking Slots: 22
19 Last Update Time: 2024-05-10 11:17:24
20
21 UID: 500105070
22 Station Name: NTNU Gongguan Campus (Student Dorm 2)
23 Available Bikes: 8
24 Available Parking Slots: 26
25 Last Update Time: 2024-05-10 11:18:24

```

The second one:

```

1 $ ./hw0402 -N "Gongguan Campus"
2 (show at most 5 stations with the substring "Gongguan Campus")
3
4 3 station(s) found with the substring "Gongguan Campus" in snaen and aren:
5
6 UID: 500105015

```

```

7 Station Name: YouBike2.0_National Taiwan Normal University Gongguan Campus
8 Available Bikes: 14
9 Available Parking Slots: 20
10 Last Update Time: 2024-05-10 11:16:24
11
12 UID: 500105061
13 Station Name: National Taiwan Normal University Gongguan Campus (Tingzhou Rd.)
14 Available Bikes: 12
15 Available Parking Slots: 22
16 Last Update Time: 2024-05-10 11:17:24
17
18 UID: 500105070
19 Station Name: NTNU Gongguan Campus (Student Dorm 2)
20 Available Bikes: 8
21 Available Parking Slots: 26
22 Last Update Time: 2024-05-10 11:18:24

```

Here are another examples:

```

1 $ ./hw0402 -U 500105061
2 (show at most 5 stations status with uid substring `500105061`.)
3 (in this case, the only station is "YouBike2.0_National Taiwan Normal
   University Gongguan Campus_1")
4
5 $ ./hw0402 -lat 25.00753 -lon 121.53719
6 (show the 5 nearest stations to the latitude and longitude)
7
8 $ ./hw0402 -U 123456
9 Error: No such station found with UID 123456

```

## 2.6 Reminder:

- You can use alternative approach to get the data, such as:  
<https://api.citybik.es/v2/networks/youbike-new-taipei>
- TA will install **libcurl** and **jq** for you to use. If you want to use other tools, please provide the installation guide in `hw0402_README`.
- Please be sure that the program can run in Ubuntu 20.04 LTS.
- The max score of this session will only be up to 30.
- You have good peers. Feel free to discuss this on the Forum or with your teammates.

## 3 Do Challenge Compiler in This Class!!(40 pts)

Professor Chi told us that when we get compiler error, we should check our code. However, why not change the compiler's code? In this problem, I want you to modify the compiler's code and add some interesting feature.

In C++, there is a well-known syntax sugar called **range-based loop**. Please reference the following site. Here is a example you can understand without the knowledge base of C++.

<https://en.cppreference.com/w/cpp/language/range-for>

In this problem, the only thing you have to do is to implement this features in C under the compiler **chibicc**. Notice that in C, there is no interfaces or polymorphism, so you only need to handle arrays. Here is an example.

<https://github.com/rui314/chibicc>

```
1 // Output:
2 // 1
3 // 2
4 // 3
5 // 4
6 // 5
7 int main()
8 {
9     int arr[5] = {1, 2, 3, 4, 5};
10    for (int x : arr)
11    {
12        printf("%d\n", arr);
13    }
14 }
```

### 3.1 Something you don't have to consider

- double, triple ... nested loops, we only have loop for 1 level.
- only based type (short, long, double, pointer).
- no init-statement.
- You can make static function non-static by yourself.
- no `__begin` and `__end` redefined inside loop

### 3.2 Testcases

I will change the size and type to prevent cheating on me. When you finish these test case, you will get at least 50% of this problem.

```
1 // Output:
2 // 1
3 // 2
4 // 3
5 // 4
6 // 5
7 int main()
8 {
9     int arr[5] = {1, 2, 3, 4, 5};
10    for (int x : arr)
```

```

11 {
12     printf("%d\n", arr);
13 }
14 }

1 // Output:
2 // 1
3 // 2
4 // 3
5 // 4
6 // 5
7 // a
8 // a
9 // a
10 // a
11 // a
12 int main()
13 {
14     char *arr[5] = {"1", "2", "3", "4", "5"};
15     for (char *x : arr)
16     {
17         printf("%s\n", x);
18         x[0] = 'a';
19     }
20     for (char *x : arr)
21     {
22         printf("%s\n", x);
23     }
24 }

```

### 3.3 How to Start

You should know how to use git before.

```

1 $ git clone https://github.com/rui314/chibicc.git
2 $ git checkout 90d1f7f # You should finish your homework under this commit
3 $ git add <all-of-the-file-you-have-modified-or-created>
4 $ git diff 90d1f7f > 1.patch # It will create 1.patch and you should submit
    this file

```

How to check if it works correctly?

```

1 $ git clone https://github.com/rui314/chibicc.git
2 $ git checkout 90d1f7f # You should finish your homework under this commit
3 $ git apply 1.patch

```

### 3.4 Hints

- `Parse.c` → `stmt`
- `codegen.c` is useless as all of the information is filled in `codegen.c`
- `preprocessor.c` is also useless



- tokenize.c → tokenize\_file

### 3.5 Some Comments

I understand many people may argue that assignment in homework of this class is not useful in career. Thus, I want to make a change on this. Compiler is a important research region in Computer Science. There are many good offers from good company for somebody who is master in compiler. In the following picture we takes Qualcomm for example.

時間不限 公司 職缺類型 資深級別 地點 薪資範圍 現增 / 選配

獲得位於美國之新Qualcomm Compiler職缺的通知。  
登入以建立職缺配對通知

在美國的 49 Qualcomm Compiler職缺

- GPU Compiler Software Development Engineer**  
美商高通國際股份有限公司  
美國 加利福尼亞州 聖克拉拉  
積極招募  
3 個月前
- GPU Compiler Performance Engineer**  
美商高通國際股份有限公司  
美國 加利福尼亞州 聖地牙哥  
積極招募  
2 個月前
- Machine Learning Compiler Engineer**  
美商高通國際股份有限公司  
美國 德克薩斯州 奧斯丁  
積極招募  
1 個月前
- GPU Compiler Software Development Engineer**  
美商高通國際股份有限公司  
美國 加利福尼亞州 聖地牙哥  
搶先應徵  
3 個月前
- Machine Learning Compiler Engineer**  
美商高通國際股份有限公司  
美國 麻塞諸塞州 波士頓  
積極招募  
1 個月前

**GPU Compiler Performance Engineer**  
美商高通國際股份有限公司 · 美國 加利福尼亞州 聖地牙哥  
2 個月前 · 39 位應徵者  
查看誰錄取了美商高通國際股份有限公司的職缺

應徵 儲存

**徵才廣告中的薪資**  
從說明中擷取。  
基本薪資範圍  
US\$ 151,000.00 / 年 - US\$ 227,000.00 / 年

**Company**  
Qualcomm Technologies, Inc.

Show more

職階等級 其他	工作類型 全職
工作職能 其他	所屬產業 電信通訊

內薦能讓您在美商高通國際股份有限公司獲得面試的機會增加 2 倍  
查看人脈

Figure 1: Qualcomm Job Opportunity.

## 4 Bonus: Wildcard (25 pts)

In this class, I have show you how to implement a process supporting lots of options and inputs. I believe that you all know how to do this. However, look the following scenario:

```
1 $ rm *.c
```

Suppose **rm** will receive only one argument **\*.c** and there is no file called **\*.c**. What happens here? Please describe how your computer works and design a lab to prove your description.

## 5 Bonus: Bit Operation (25 pts)

I want to write a program to display a 32-bit integer in the binary form. So I write the following code. However, this code has some problem.

```
1 #include <stdio.h>
2 #include <stdint.h>
3
4 int main()
5 {
6     int32_t number = 0;
7
8     scanf( "%d", & number );
9
10    int32_t bit = 1;
11    bit = bit << 31;
12
13    for( int i = 0 ; i < 32 ; i++ )
14    {
15        if( bit & number )
16            printf( "1" );
17        else
18            printf( "0" );
19        bit = bit >> 1;
20    }
21    return 0;
22 }
```

Please explain the reason of the problem of this code and show how to fix it.

## 6 Bonus: MACRO (25 pts)

Please read the following code.

```
1 #include <stdio.h>
2
3 #define Peval(cmd) printf( #cmd ": %g\n", cmd );
4
5 int main()
6 {
7     double *plist = (double[]){1, 2, 3};
```

```
8  double list[] = {1, 2, 3};
9  Peval( sizeof( plist ) / ( sizeof( double ) + 0.0 ) );
10 Peval( sizeof( list ) / ( sizeof( double ) + 0.0 ) );
11 return 0;
12 }
```

Please run this code and **explain this macro**.

## 7 Bonus: **tgmath.h** (25 pts)

In C99, there are some new standard header files. One of them is **tgmath.h**. You can find there is a "function" called **pow** declared in this file. What is the difference between **tgmath.h** and **math.h**? Please use **pow** as an example to describe your opinion.