HW3 Program

Team組員

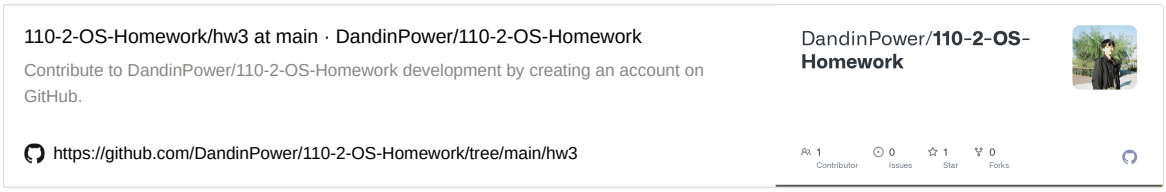
資工三 108590017 林銘政	課本範例查詢與環境架設
資工三 108590043 李永祺	虛擬機測試操作
資工三 108590045 廖永誠	程式碼邏輯撰寫

Environment

- 使用Ubuntu20.04.1
- 需有以下兩個套件

```
sudo apt-get install make
sudo apt-get install flex
```

- Github地址



Problem

7.17

- 編譯程式碼

```
gcc -o main main.c -lpthread
```

- 執行程式

```
./main
```

- 輸出結果
  - 會確保不會同時有south跟north的人上橋
  - 如果是同一邊的話則可以正常上橋

```
dandinpower123@ubuntu:~/Desktop/110-2-OS-Homework/hw3/problem/7.17$ gcc -o main main.c -lpthread
dandinpower123@ubuntu:~/Desktop/110-2-OS-Homework/hw3/problem/7.17$ ./main
[2]'s farmers from north want to across the bridge...
[2]'s farmers from north acrossing the bridge...
[1]'s farmers from north want to across the bridge...
[1]'s farmers from north acrossing the bridge...
[3]'s farmers from south want to across the bridge...
[1]'s farmers from south want to across the bridge...
[0]'s farmers from south want to across the bridge...
[2]'s farmers from south want to across the bridge...
[0]'s farmers from north want to across the bridge...
[0]'s farmers from north acrossing the bridge...
[3]'s farmers from north want to across the bridge...
[3]'s farmers from north acrossing the bridge...
[1]'s farmers from north leaving the bridge...
[4]'s farmers from south want to across the bridge...
[4]'s farmers from north want to across the bridge...
[4]'s farmers from north acrossing the bridge...
[3]'s farmers from north leaving the bridge...
[2]'s farmers from north leaving the bridge...
[0]'s farmers from north leaving the bridge...
[4]'s farmers from north leaving the bridge...
The bridge is empty!
[3]'s farmers from south acrossing the bridge...
[1]'s farmers from south acrossing the bridge...
[0]'s farmers from south acrossing the bridge...
[2]'s farmers from south acrossing the bridge...
[4]'s farmers from south acrossing the bridge...
[0]'s farmers from south leaving the bridge...
[2]'s farmers from south leaving the bridge...
[4]'s farmers from south leaving the bridge...
[3]'s farmers from south leaving the bridge...
[1]'s farmers from south leaving the bridge...
The bridge is empty!
```

8.25

- 編譯程式碼

```
gcc -o main main.c
```

- 執行程式

```
./main <virtual address>
```

- 輸出結果

```
dandinpower123@ubuntu:~/Desktop/110-2-OS-Homework/hw3/problem/8.25$ gcc -o main main.c
dandinpower123@ubuntu:~/Desktop/110-2-OS-Homework/hw3/problem/8.25$ ./main 19986
The address 19986 contains:
page number = 4
offset = 3602
```

9.26

- 編譯程式碼

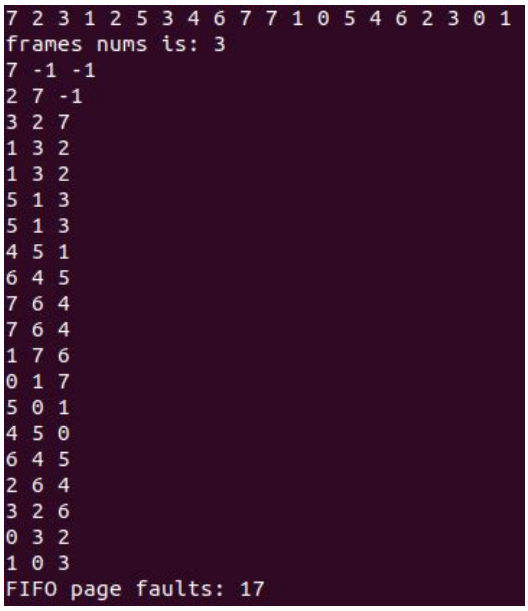
```
gcc -o main main.c
```

- 執行程式

```
./main <frameNums>
```

  - frameNums 為frame的個數

- 輸出說明
  1. 先根據handwrite 9.8的題目當測資
  2. 再隨機產生一組測資
- 輸出結果
  - handwrite 9.8



```
7 -1 -1
2 7 -1
3 2 7
1 3 2
2 1 3
5 2 1
3 5 2
4 3 5
6 4 3
7 6 4
7 6 4
1 7 6
0 1 7
5 0 1
4 5 0
6 4 5
2 6 4
3 2 6
0 3 2
1 0 3
LRU page faults: 18
7 -1 -1
7 2 -1
7 2 3
1 2 3
1 2 3
5 1 3
5 1 3
4 5 1
6 5 1
7 5 1
7 5 1
7 5 1
0 5 1
0 5 1
4 0 1
6 0 1
2 0 1
3 0 1
3 0 1
3 0 1
OPA page faults: 13
```

- Random generate

```
Random Generate Sequence:
0 7 2 3 3 9 7 4 0 8 5 7 7 8 9 3 8 9 4 8
frames nums is: 3
0 -1 -1
7 0 -1
2 7 0
3 2 7
3 2 7
9 3 2
7 9 3
4 7 9
0 4 7
8 0 4
5 8 0
7 5 8
7 5 8
7 5 8
9 7 5
3 9 7
8 3 9
8 3 9
4 8 3
4 8 3
FIFO page faults: 15
```

Project

Chap7

- 專案說明
  1. 選用了5個customers,以及4種resources
  2. 先透過argv宣告好resources的available
  3. maximum會根據available來隨機產生(每個max都不會超過available)
  4. create 5個customer thread來不斷request resources並且在need 都為0後release所有resource
  5. 當全部的need都配置完後及結束程式
- 輸出說明
  1. request resource時
     - request的值為根據該customer的need來隨機產生

     ```
     [編號]customer is request: <1> <2> <3> <4>
     ```

     - 如果成功的話

```
Sequence: [找出符合safe state的執行順序] Request is accept!
```

- 如果失敗的話

```
Sequence: -1 Request is Reject and wait for resource because can't find valid sequence,so it's in unsafe state.
```

2. 當都配置完後release all

```
[編號]customer is finish!
[編號]customer release all!
```

3. 每當available,maximum,allocation,need有更新時都會輸出一次目前的狀態

```
Available:
3 1 6 0

Maximum:
8 9 7 2
7 5 3 7
6 2 4 2
3 5 8 1
6 5 8 0

Allocation:
5 8 1 2
0 0 1 5
0 0 0 0
0 0 0 0
2 0 0 0

Need:
3 1 6 0
7 5 2 2
0 0 0 0
3 5 8 1
4 5 8 0
```

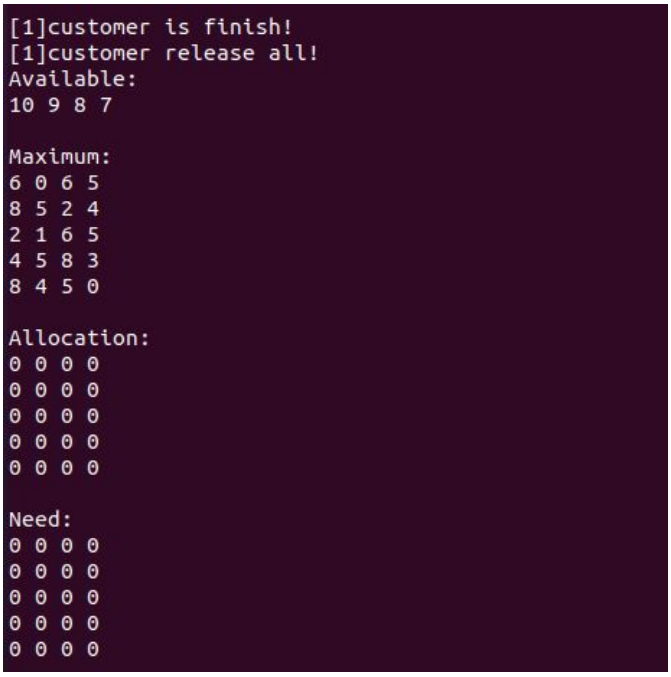
- 編譯程式碼

```
gcc -o main main.c -lpthread
```

- 執行程式

```
./main <available0> <available1> <available2> <available3>
```

- 輸出結果



- 此為最終的結束狀態 → 上面省略

Chap9

- 專案說明
  - 採用了3個檔案
    - addresses.txt 為測資1000筆
    - output.txt 為紀錄output輸出
    - BACKING_STORE.bin 為長度為65536長度的memory資料
  - 在執行的時候透過指定argv來把以上3個檔案的路徑輸入進去
- 輸出說明
  - 輸出每一行的logical address,physical address,還有對應的value
  - 輸出測資的筆數
  - 輸出TLB hits的個數以及比例
  - 輸出Page faults的個數以及比例
- 編譯程式碼

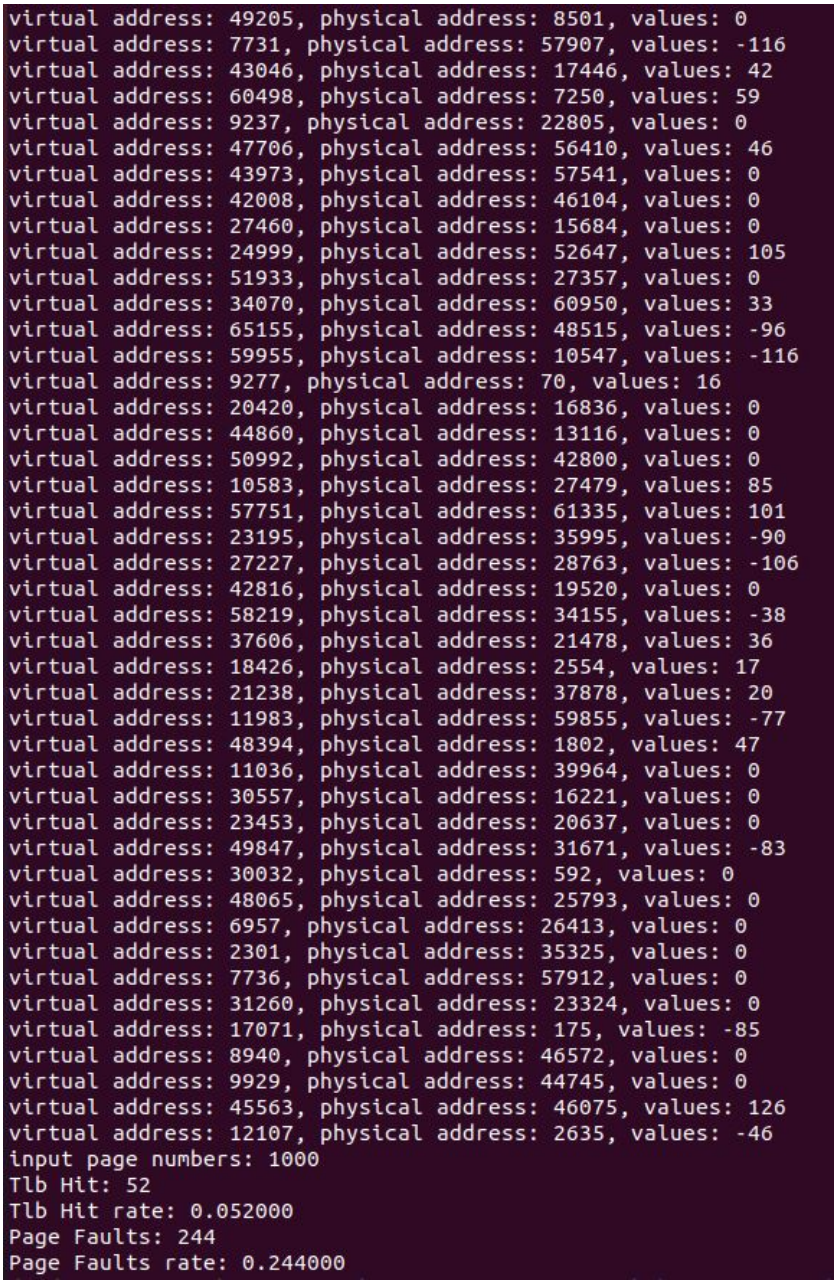
```
gcc -o main main.c
```

- 執行程式碼

```
./main BACKING_STORE.bin addresses.txt output.txt
```

- 輸出結果

```
cat output.txt
```


virtual address: 49205, physical address: 8501, values: 0
virtual address: 7731, physical address: 57907, values: -116
virtual address: 43046, physical address: 17446, values: 42
virtual address: 60498, physical address: 7250, values: 59
virtual address: 9237, physical address: 22805, values: 0
virtual address: 47706, physical address: 56410, values: 46
virtual address: 43973, physical address: 57541, values: 0
virtual address: 42008, physical address: 46104, values: 0
virtual address: 27460, physical address: 15684, values: 0
virtual address: 24999, physical address: 52647, values: 105
virtual address: 51933, physical address: 27357, values: 0
virtual address: 34070, physical address: 60950, values: 33
virtual address: 65155, physical address: 48515, values: -96
virtual address: 59955, physical address: 10547, values: -116
virtual address: 9277, physical address: 70, values: 16
virtual address: 20420, physical address: 16836, values: 0
virtual address: 44860, physical address: 13116, values: 0
virtual address: 50992, physical address: 42800, values: 0
virtual address: 10583, physical address: 27479, values: 85
virtual address: 57751, physical address: 61335, values: 101
virtual address: 23195, physical address: 35995, values: -90
virtual address: 27227, physical address: 28763, values: -106
virtual address: 42816, physical address: 19520, values: 0
virtual address: 58219, physical address: 34155, values: -38
virtual address: 37606, physical address: 21478, values: 36
virtual address: 18426, physical address: 2554, values: 17
virtual address: 21238, physical address: 37878, values: 20
virtual address: 11983, physical address: 59855, values: -77
virtual address: 48394, physical address: 1802, values: 47
virtual address: 11036, physical address: 39964, values: 0
virtual address: 30557, physical address: 16221, values: 0
virtual address: 23453, physical address: 20637, values: 0
virtual address: 49847, physical address: 31671, values: -83
virtual address: 30032, physical address: 592, values: 0
virtual address: 48065, physical address: 25793, values: 0
virtual address: 6957, physical address: 26413, values: 0
virtual address: 2301, physical address: 35325, values: 0
virtual address: 7736, physical address: 57912, values: 0
virtual address: 31260, physical address: 23324, values: 0
virtual address: 17071, physical address: 175, values: -85
virtual address: 8940, physical address: 46572, values: 0
virtual address: 9929, physical address: 44745, values: 0
virtual address: 45563, physical address: 46075, values: 126
virtual address: 12107, physical address: 2635, values: -46
input page numbers: 1000
Tlb Hit: 52
Tlb Hit rate: 0.052000
Page Faults: 244
Page Faults rate: 0.244000

- 以上省略