

#404410082 資工三 李宗樺

##題目

## 作業內容

### 第一部分

- 設計一個實驗，了解作業系統配置記憶體的方法
- 經由這個實驗回答：malloc 40MB 時，作業系統是否會立刻配置 40MB 的記憶體給該程式
- 了解作業系統是需求分頁，當你跟作業系統要求配置某個數量的記憶體，實際上作業系統不會馬上給你，除非真的使用該快記憶體，例如對記憶體內每一個 byte 進行讀寫

### 第二部分

- 撰寫一支程式可要求大量的 memory，迫使 system 做 swap，之後使用 free()使 used memory 變成 free memory，由於曾經 swap 過因此最後的 free memory 會比之前多。
- 了解作業系統是以全域的方式配置記憶體，當跟系統要很多記憶體，就會將其他 process 的記憶體『擠出去』到 swap space
- 可輸入一個參數 int，表示要求多少 GB 的 memory。例如：  
./free\_mem 2，表示 2\*1024\*1024\*1024 byte。
- 必須提供觀察方式，但不可使用 GUI、系統監控（相當於 Windows 的系統管理員）。

##第一個實驗

###說明你實驗的依據

#### 1. malloc 40MB

```
int *mem = (int*)malloc(sizeof(int)*10*1024*1024);
```

#### 2. 對每一個位置做存取

```
memset(mem, 0, sizeof(int)* 10* 1024 * 1024);
```

#### 3. 用 command "top -p pid" 觀察

###說明你所設計的程式是否有特別之處

#### 4. 用getpid()得到process id，方便觀察。

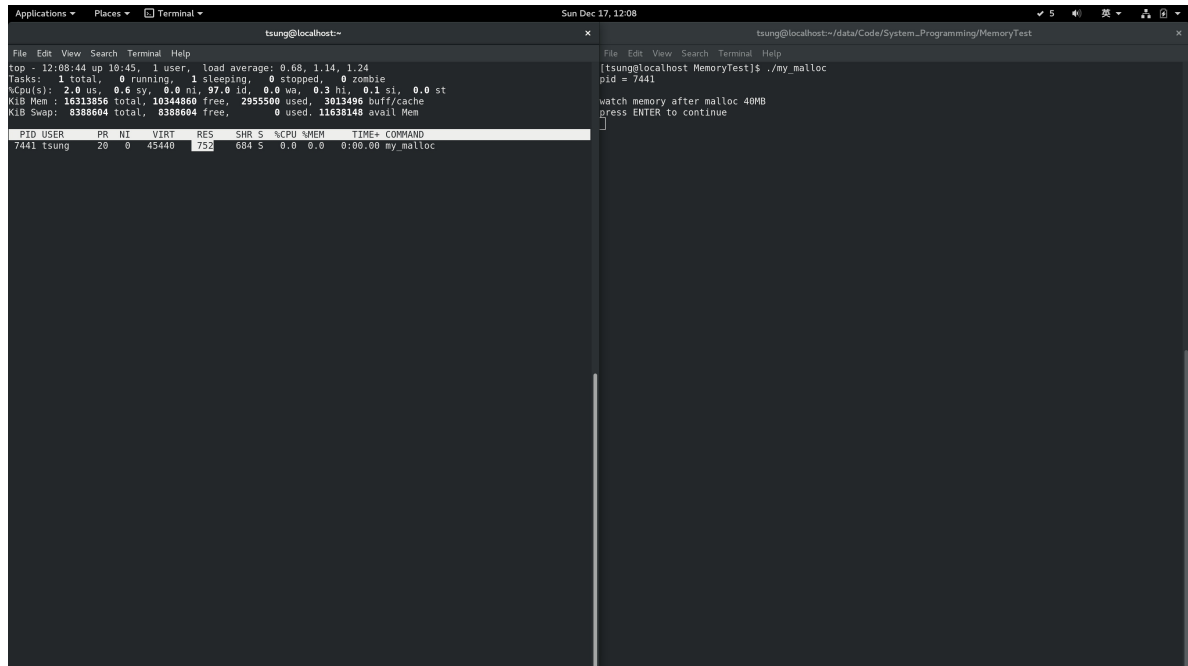
#### 5. 記憶體存取直接使用memset()。

#### 6. 用getchar()等待，方便觀察。

###你觀察 Linux 中那些檔案或者資訊，以驗證你的論述

用 top 指令，觀察process的RES(anything occupying physical memory)。

觀察只有malloc，卻還沒對記憶體操作時，RES 只有 752 KB。



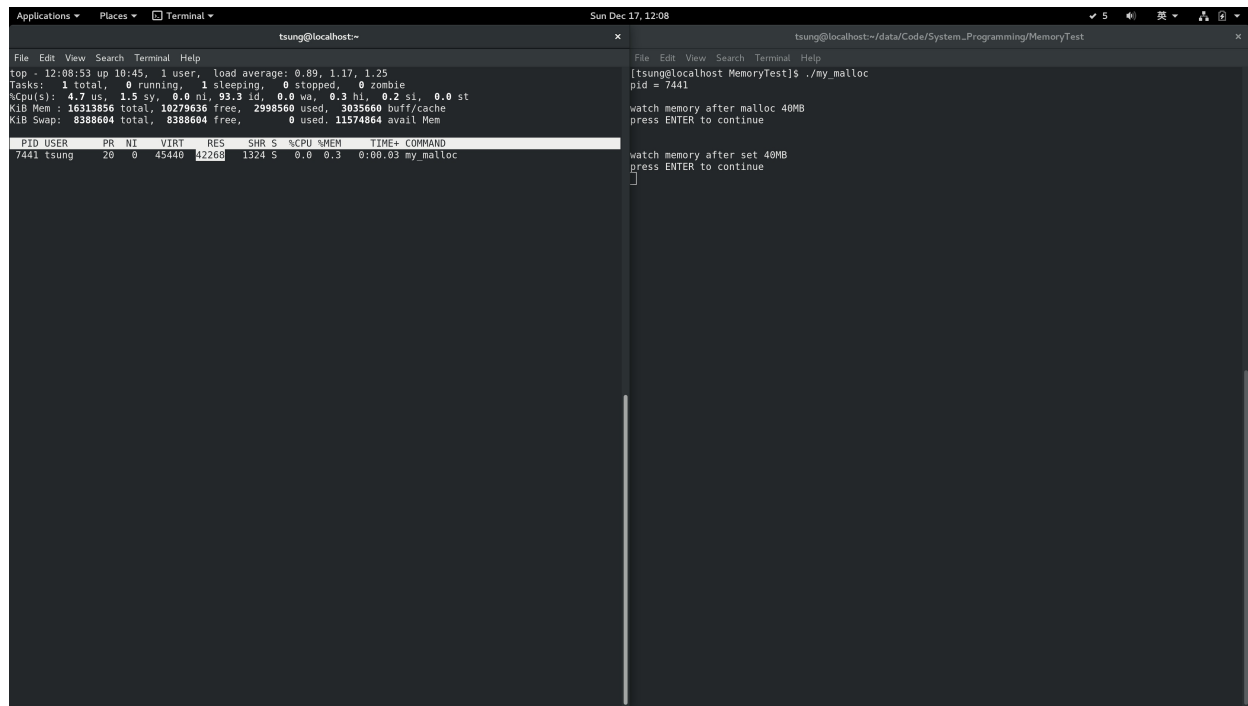
The screenshot shows two terminal windows. The left window displays the output of the 'top' command, showing system metrics and a table of processes. The right window shows the execution of a program named 'my\_malloc' which is waiting for user input.

```
top - 12:08:44 up 10:45, 1 user, load average: 0.68, 1.14, 1.24
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.0 us, 0.6 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.3 hi, 0.1 si, 0.0 st
KiB Mem : 16313856 total, 10344060 free, 2955500 used, 3013496 buff/cache
KiB Swap: 8388604 total, 8388604 free, 0 used, 11638148 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7441	tsung	20	0	45440	752	884	S	0.0	0.0	0:00.00	my_malloc

```
[tsung@localhost MemoryTest]$ ./my_malloc
pid = 7441
watch memory after malloc 40MB
press ENTER to continue
```

觀察對每個記憶體做存取後，RES 42268 KB。



The screenshot shows the same two terminal windows as before, but now the 'top' command shows that the 'my\_malloc' process has increased its memory usage significantly.

```
top - 12:08:53 up 10:45, 1 user, load average: 0.89, 1.17, 1.25
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.7 us, 1.5 sy, 0.0 ni, 93.3 id, 0.0 wa, 0.3 hi, 0.2 si, 0.0 st
KiB Mem : 16313856 total, 10278636 free, 2999560 used, 3035660 buff/cache
KiB Swap: 8388604 total, 8388604 free, 0 used, 11574664 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7441	tsung	20	0	45440	42268	1324	S	0.0	0.3	0:00.03	my_malloc

```
[tsung@localhost MemoryTest]$ ./my_malloc
pid = 7441
watch memory after malloc 40MB
press ENTER to continue

[tsung@localhost MemoryTest]$ ./my_malloc
pid = 7441
watch memory after set 40MB
press ENTER to continue
```

##第二個實驗

###說明你的程式花費了多少時間，釋放了多少記憶體

```
tsung@localhost:~/data/Code/System_Programming/MemoryTest
File Edit View Search Terminal Help
[tsung@localhost MemoryTest]$ time ./free_mem 13
4220
程式開始前,用top/free指令觀察
press ENTER to continue

配置大量記憶體,用top/free指令觀察
press ENTER to continue

real    0m4.446s
user    0m0.581s
sys     0m3.371s
[tsung@localhost MemoryTest]$
```

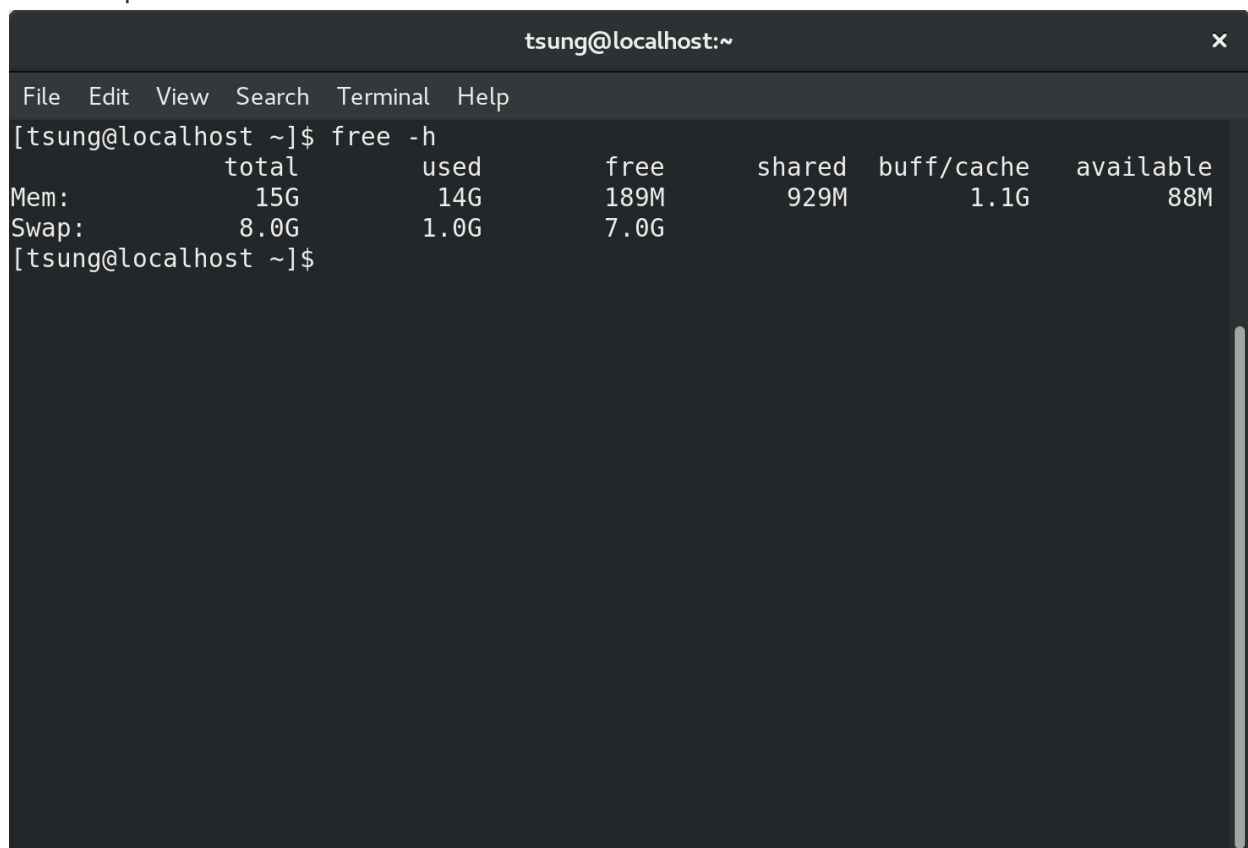
###說明你的 free\_mem 對於 swap space 的影響（即：是否額外造成 page-out 及 page-in），如何觀察？

在程式開始前，先用free -h 觀察資訊，發現Mem free 11G。

```
tsung@localhost:~
File Edit View Search Terminal Help
[tsung@localhost ~]$ free -h
              total        used        free      shared  buff/cache   available
Mem:          15G         1.8G         11G         1.3G         2.3G         12G
Swap:          8.0G           0B         8.0G
[tsung@localhost ~]$
```

開始執行程式，宣告大量的記憶體 malloc 14G，，迫使 system 做 swap。

發現Swap used 1GB。

A terminal window titled 'tsung@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'free -h' has been executed, displaying memory usage statistics. The output shows that memory is mostly free (13G), but swap space has 1.0G used and 7.0G free. The terminal has a scrollbar on the right side.

```
tsung@localhost ~]$ free -h
```

	total	used	free	shared	buff/cache	available
Mem:	15G	14G	189M	929M	1.1G	88M
Swap:	8.0G	1.0G	7.0G			

```
tsung@localhost ~]$
```

程式結束後, Mem free 13G, 比值前多。

Swap used 依然是1G

```
tsung@localhost:~  
File Edit View Search Terminal Help  
[tsung@localhost ~]$ free -h  
              total        used        free      shared  buff/cache   available  
Mem:           15G         1.2G         13G          929M         1.1G         13G  
Swap:          8.0G         1.0G         7.0G  
[tsung@localhost ~]$
```