

Homework Assignment No. 5

土木所電輔組碩一 R07521603 蔡松霖

從這次的作業可以體悟到把memory釋放的重要性。第一題中每次執行呼叫1000,000,000 個integer的儲存空間時，會要走系統上90%左右的記憶體資源(如圖 2所示)。多執行幾個步驟以後，會發現程式執行的越來越慢，同時%MEM一直保持在90%左右，且在沒有釋放memory的情況下會因為內存不夠用而中止程式的運作(core dumped，如圖 3所示)。而在加入執行freeArray()函式下釋放記憶體，可以觀察到雖然每次執行完一次一樣會佔用到約90%的記憶體，但是再執行下一次前會釋放掉先前佔有的部分，%MEM會先歸零再隨著執行步驟上升記憶體的使用。也能因此一直執行程式下去。第二題使用shared pointer來儲存陣列，不需要自己寫free memory的函式，即可達到跟上一題有釋放記憶體的程式一樣的效果，當指向該reference counter=0時，就會把該reference佔有的空間釋放掉。

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1100	lightdm	20	0	45248	3000	2688	S	0.0	0.1	0:01.18	systemd
26679	root	20	0	274948	2968	2628	S	0.0	0.1	0:00.21	cups-browsed
1144	lightdm	20	0	178532	2468	2224	S	0.0	0.1	1:08.72	dconf-service
1223	root	20	0	347480	2364	2140	S	0.0	0.1	0:02.33	upowerd
936	root	20	0	337384	2284	2020	S	0.0	0.1	0:05.44	ModemManager
955	root	20	0	28632	2196	2004	S	0.0	0.1	0:15.01	systemd-logind
5542	tslsun	20	0	12880	2148	1988	S	0.0	0.1	0:00.00	sftp-server
928	avahi	20	0	45752	2064	1204	S	0.0	0.1	4:52.19	avahi-daemon
3780	bordenc+	20	0	12880	2028	1872	S	0.0	0.1	0:00.00	sftp-server
1328	whoopsie	20	0	378496	1788	1588	S	0.0	0.0	0:05.42	whoopsie
5308	tslsun	20	0	210824	1784	0	S	0.0	0.0	0:00.00	(sd-pam)
045	root	20	0	166744	1772	1500	S	0.0	0.0	1:14.15	thunderbird
6410	tslsun	20	0	13300	1696	1552	S	0.0	0.0	0:00.00	a.out
5727	bordenc+	20	0	210824	1676	0	S	0.0	0.0	0:00.00	(sd-pam)
1113	lightdm	20	0	42980	1424	1072	S	0.0	0.0	0:38.71	dbus-daemon
1065	root	20	0	65508	1372	1260	S	0.0	0.0	0:04.18	sshd
1207	lightdm	9	-11	416480	1272	1108	S	0.0	0.0	0:02.64	pulseaudio
925	root	20	0	29288	1148	1052	S	0.0	0.0	0:03.18	cron
1137	lightdm	20	0	206856	848	796	S	0.0	0.0	0:37.75	at-spi2-registr

圖 1 初始狀態

PTD	USER	PR	NT	VRT	RFS	SHR	S	%CPU	%MEM	TIME+	COMMAND
6410	tslsun	20	0	3919552	3.390g	140	S	0.0	90.8	0:09.08	a.out
6424	tslsun	20	0	42000	320	220	R	0.0	0.1	0:00.01	top
1156	lightdm	20	0	621372	2576	2104	S	0.0	0.1	1:08.18	unity-settings-
304	root	20	0	32084	2492	2360	S	0.0	0.1	0:32.18	systemd-journal
6121	tslsun	20	0	22708	2428	2008	S	0.0	0.1	0:00.05	bash
928	avahi	20	0	45752	1732	992	S	0.0	0.0	4:52.19	avahi-daemon
1163	lightdm	20	0	679244	752	4	S	0.0	0.0	0:36.72	indicator-sound
1053	root	20	0	352012	720	284	S	0.0	0.0	8:35.10	Xorg
1158	lightdm	20	0	373440	720	0	S	0.0	0.0	0:35.49	indicator-messa
1114	lightdm	20	0	987028	668	0	S	0.0	0.0	3:47.16	unity-greeter
1162	lightdm	20	0	658864	528	0	S	0.0	0.0	13:46.73	indicator-keybo
954	root	20	0	293228	424	0	S	0.0	0.0	2:36.63	accounts-daemon
1048	root	20	0	19600	240	172	S	0.0	0.0	1:23.50	irqbalance

圖 2 執行一次步驟後所占記憶體

```
tslsun@cp2r-M32AA-Invalid-entry-length-16-Fixed-up-to-11:~/oop/files/HW5/HW5_1$ ./a.out
How many elements do you want in your array? 1000000000
How many elements do you want in your array? 1000000000
How many elements do you want in your array? 1000000000
How many elements do you want in your array? 1000000000
terminate called after throwing an instance of 'std::bad_alloc'
  what():  std::bad_alloc
Aborted (core dumped)
```

圖 3 第四次的呼叫時內存就用盡了