

Subject: Lecture on Operating System Lecture Exercise 10.2

From: IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

Date: Wed, 24 Jan 2007 18:18:42 +0900

To: Shinji KONO <kono@ie.u-ryukyu.ac.jp>

学籍番号 : 055702B

問題10.2

メモリを大量に消費するプログラムを C または、Perl で記述し、実際に swap が起きる様子を観察せよ。

- ・ソースの抜粋

Operating System Lecture No.8の malloc_test.c を使用した。
ACTIVE=10000000に変更した。

ソースの場所 : /home/y05/j05002/OS-Lecture10/Lec10-2/malloc_test.c

```
#include <stdio.h>
#include <stdlib.h>
```

```
int COUNT = 80000;
int ACTIVE = 1000000000;
int MAXSIZE = 4000;
int MINSIZE = 400;
int MIN_UNIT = 1 ;
```

省略

```
int
main(int ac, char *av[])
{
MEM_LIST mlist;
MEM_LIST_PTR last = &mlist;
MEM_LIST_PTR new;
int i,size;

option(ac,av);

mlist.address = NULL;
mlist.size = 0;
for(i=0;i<ACTIVE;i++) {
new = (MEM_LIST_PTR)malloc(sizeof(MEM_LIST));
if (!new) die("malloc error");
new->address = NULL;
new->next = NULL;
last->next = new;
last = new;
}
last->next = &mlist;

for(i=0;i++<COUNT;last=last->next) {
size = ((random()%(MAXSIZE-MINSIZE))+MINSIZE)*MIN_UNIT;
if (last->address) {
free(last->address);
}
last->size = size;
last->address = (void *)malloc(size);
if (!last->address) die("malloc error");
}
print_mem_list(&mlist);
```

}

・実行結果

```
[j05002@pw002 Lec10-2]% ./malloc_test  
強制終了  
[j05002@pw002 Lec10-2]%
```

別ウィンドウで実行した、topコマンドの結果を以下に示す。

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  
6337 j05002 20 0 16624 16m 216 R 7.9 3.2 0:00.24 malloc_test  
6336 j05002 12 0 984 980 748 R 0.7 0.2 0:00.37 top  
1 root 8 0 268 236 220 S 0.0 0.0 0:24.75 init  
2 root 9 0 0 0 0 S 0.0 0.0 0:00.45 keventd  
3 root 9 0 0 0 0 S 0.0 0.0 0:00.00 kapmd  
4 root 19 19 0 0 0 S 0.0 0.0 0:00.11 ksoftirqd_CPU0  
5 root 9 0 0 0 0 S 0.0 0.0 2:53.72 kswapd  
6 root 9 0 0 0 0 S 0.0 0.0 0:00.00 bdflush
```

数秒経過した時の結果

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  
6272 j05002 20 0 105m 105m 216 R 50.0 20.9 0:01.51 malloc_test  
5 root 20 0 0 0 0 R 4.0 0.0 2:51.83 kswapd  
6271 j05002 11 0 988 984 752 R 0.7 0.2 0:00.12 top  
6065 j05002 9 0 1964 1960 1128 S 0.3 0.4 0:00.20 tcsh
```

topコマンドの結果より、malloc_testを実行し、ある程度メモリの使用量が大きくなると、最初スリープ状態にあったkswapdが実行状態に遷移していることがわかる。

kswapdはスワップデーモンであり、フリーメモリ量を監視しており、一定量を切ると実行される。