CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 05: Virtual Memory

Rahmat M. Samik-Ibrahim

University of Indonesia

http://rms46.vlsm.org/2/207.html
Always check for the latest revision!

REV115 05-FEB-2018

Operating Systems 2018-1 (Room 3114 Tue/Thu)
Class: A (10:00-12:00) | B (13:00-15:00) | C (16:00-18:00)

Week	Schedule	Topic	OSC9
Week 00	06 Feb - 12 Feb 2018	Intro & Review1	Ch. 1, 16
Week 01	13 Feb - 19 Feb 2018	Review2 & Scripting	Ch. 1, 2
Week 02	20 Feb - 26 Feb 2018	Protection, Security, Privacy, & C-language	Ch. 14, 15
Week 03	27 Feb - 05 Mar 2018	I/O, BIOS, Loader, & Systemd	Ch. 13
Week 04	06 Mar - 12 Mar 2018	Addressing, Shared Lib, & Pointer	Ch. 8
Week 05	13 Mar - 19 Mar 2018	Virtual Memory	Ch. 9
Reserved	20 Mar - 24 Mar 2018		
Mid-Term	26 Mar - 03 Apr 2018	(UTS)	
Week 06	05 Apr - 11 Apr 2018	Concurency: Processes & Threads	Ch. 3, 4
Week 07	12 Apr - 18 Apr 2018	Synchronization	Ch. 5, 7
Week 08	19 Apr - 25 Apr 2018	Scheduling	Ch. 6
Week 09	26 Apr - 05 May 2018	File System & Persistent Storage	Ch. 10, 11, 12
Week 10	07 May - 16 May 2018	I/O Programming & Network Sockets Programming	
Reserved	17 May - 22 May 2018		
Final	23 May - 26 May 2018	(UAS)	
Deadline	07 Jun 2018 16:00	Extra assignment **deadline**	

Week 05: Memory

Memory

- Reference: (OSCE2e ch7/8) (UCB 11 12 13) (UDA P3L2) (OLD 06)
- Binding & Linking
  - Address Binding
  - Address Space: Logical & Physical
  - Dynamic & Static Linking
  - MMU: Memory Management Unit
  - Base and Limit Registers
  - Swapping
  - Mobile Systems Problem: no swap
- Memory Allocation
  - Contiguous Allocation
  - Multiple-variable-partition Allocation
  - First, Best, Worst Fit Allocation Strategy
- Fragmentation
  - External
  - Internal
  - Compaction

Paging

- Address Space
- Logical/Virtual Address
- Pages
- Page Number
- Page Offset
- Page Table
- PTE: Page Table Entry
- Page Flags: Valid/ Invalid
- TLBs: Translation Look-aside Buffers/ Associative Memory
- Physical Address
- Frames

Address Translation Scheme

Address		Binary								
DEC	HEX	OFFSET	PG	OFF	PG	OFF	PAGE	OFF	PAGE	OFF
00	00	00000	0	0000	00	000	000	00	0000	0
01	01	00001	0	0001	00	001	000	01	0000	1
02	02	00010	0	0010	00	010	000	10	0001	0
03	03	00011	0	0011	00	011	000	11	0001	1
04	04	00100	0	0100	00	100	001	00	0010	0
05	05	00101	0	0101	00	101	001	01	0010	1
06	06	00110	0	0110	00	110	001	10	0011	0
07	07	00111	0	0111	00	111	001	11	0011	1
08	08	01000	0	1000	01	000	010	00	0100	0
09	09	01001	0	1001	01	001	010	01	0100	1
10	0A	01010	0	1010	01	010	010	10	0101	0
11	0B	01011	0	1011	01	011	010	11	0101	1
12	0C	01100	0	1100	01	100	011	00	0110	0
13	0D	01101	0	1101	01	101	011	01	0110	1
14	0E	01110	0	1110	01	110	011	10	0111	0
15	0F	01111	0	1111	01	111	011	11	0111	1
16	10	10000	1	0000	10	000	100	00	1000	0
17	11	10001	1	0001	10	001	100	01	1000	1
18	12	10010	1	0010	10	010	100	10	1001	0
19	13	10011	1	0011	10	011	100	11	1001	1
20	14	10100	1	0100	10	100	101	00	1010	0
21	15	10101	1	0101	10	101	101	01	1010	1
22	16	10110	1	0110	10	110	101	10	1011	0
23	17	10111	1	0111	10	111	101	11	1011	1
24	18	11000	1	1000	11	000	110	00	1100	0
25	19	11001	1	1001	11	001	110	01	1100	1
26	1A	11010	1	1010	11	010	110	10	1101	0
27	1B	11011	1	1011	11	011	110	11	1101	1
28	1C	11100	1	1100	11	100	111	00	1110	0
29	1D	11101	1	1101	11	101	111	01	1110	1
30	1E	11110	1	1110	11	110	111	10	1111	0
31	1F	11111	1	1111	11	111	111	11	1111	1

Memory (20 bits)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
00010	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
00020	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
00030	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
FFFF0																

Hierarchical Page Table

- OPT: outer page table (P1)
- PT: page table (P2)
- Offset (D)
- Three-level Paging Scheme
- Hashed Page Tables
- Inverted Page Table
- Demand Paging
- Copy On Write (COW)

VM

- Page Replacement Algorithm
  - Reference String
  - First-In-First-Out (FIFO)
  - Belady Anomaly
  - Optimal Algorithm
  - Least Recently Used (LRU)
  - LRU Implementation
  - Lease Frequently Used (LFU)
  - Most Frequently Used (MFU)
- Frame Allocation
- Global vs. Local Allocation
- Non-Uniform Memory Access (NUMA)
- Working-Set Model
- Kernel
  - Buddy System Allocator
  - Slab Allocator

TOP

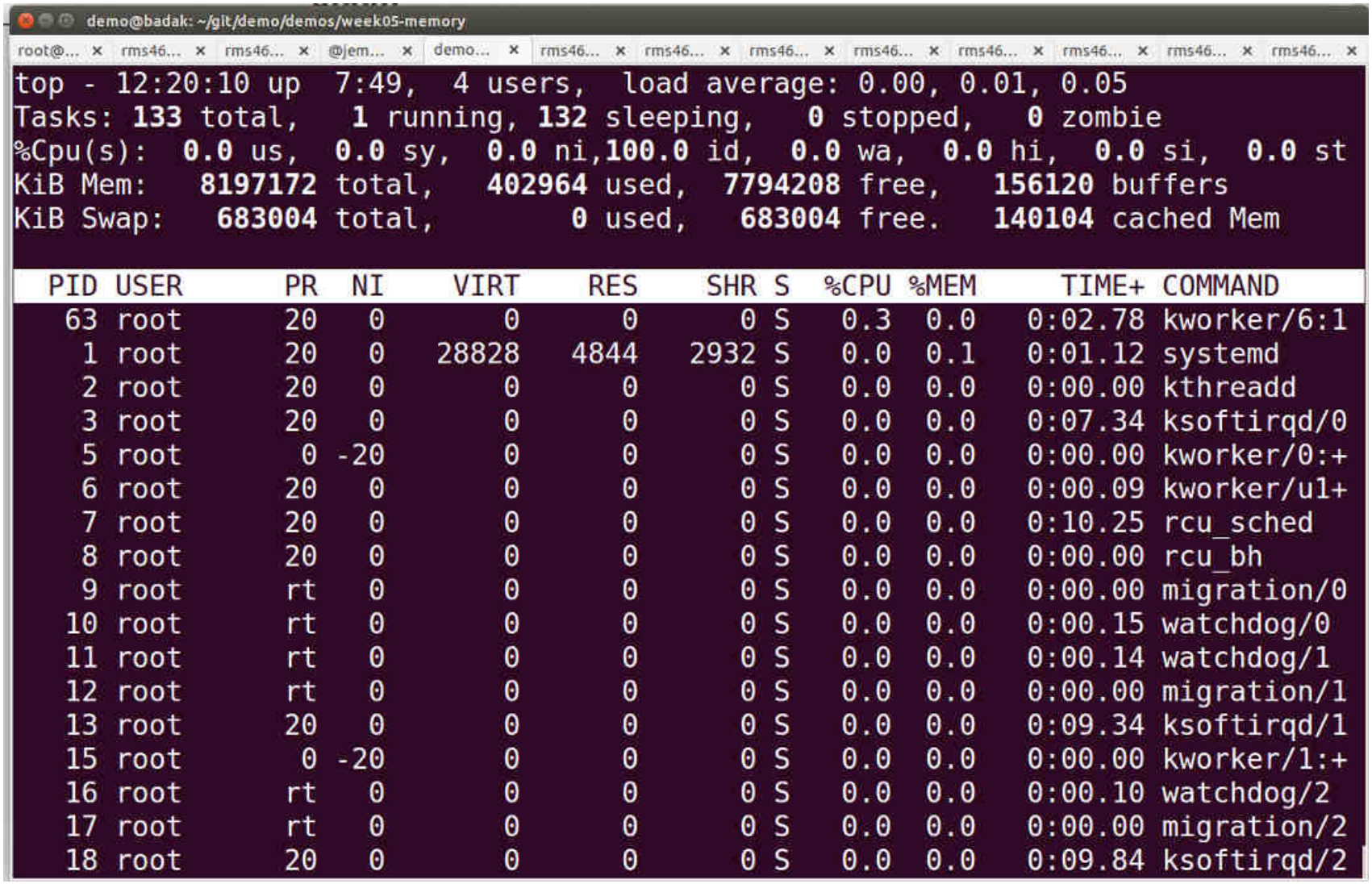

Figure: top

TOP (2)

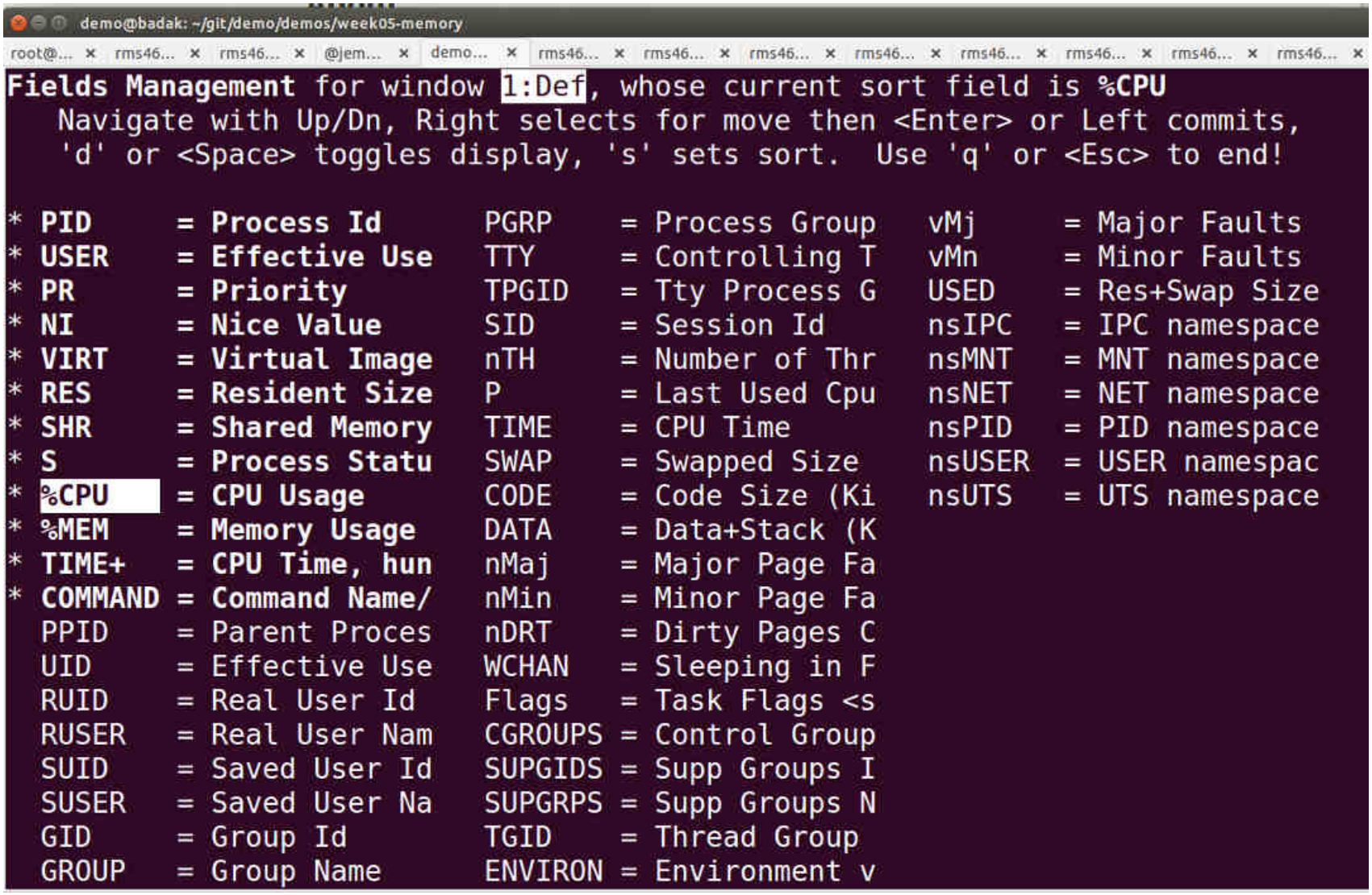

Figure: "h" = help

TOP (3)

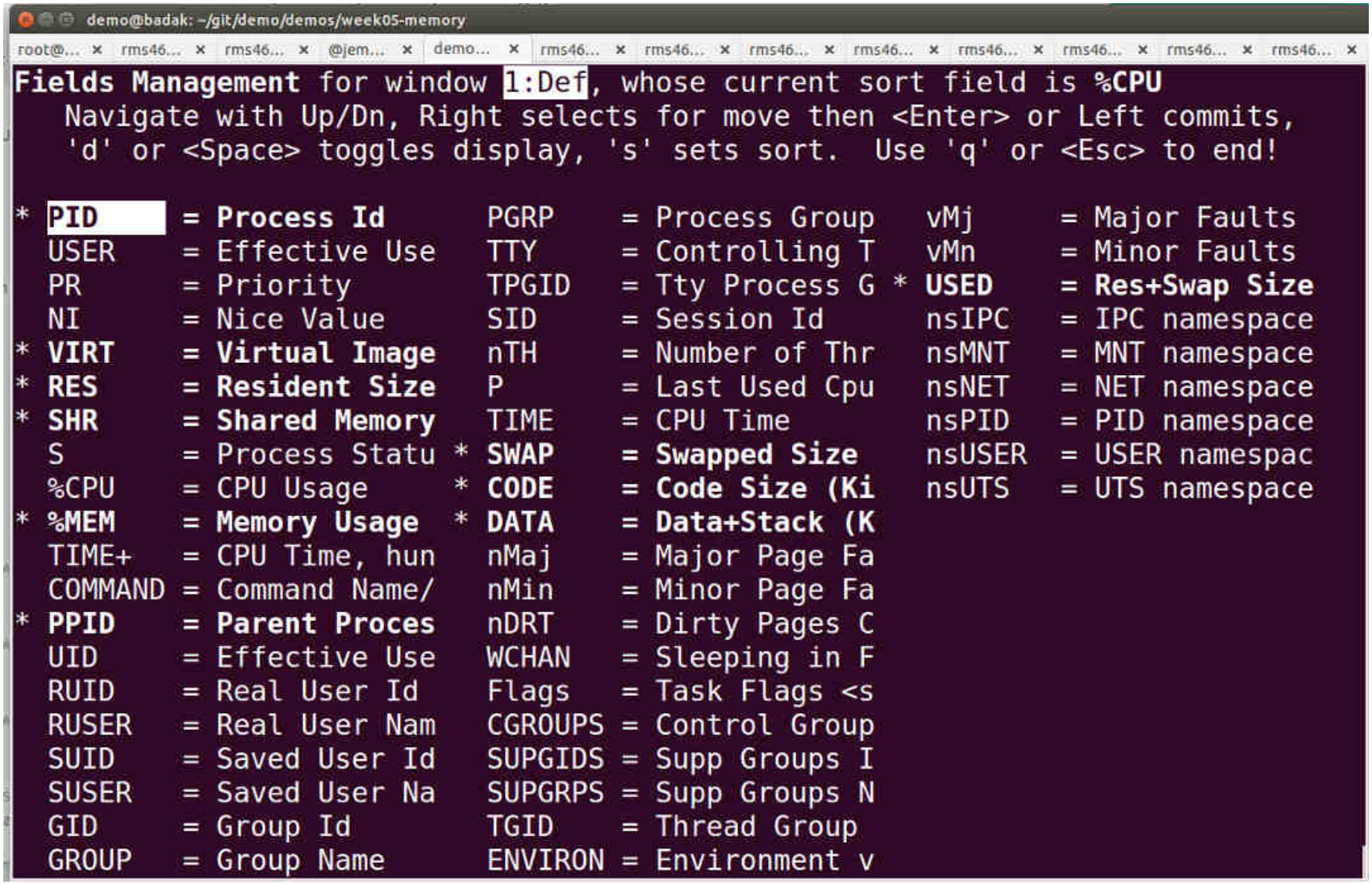

Figure: Moving Fields

TOP (4)

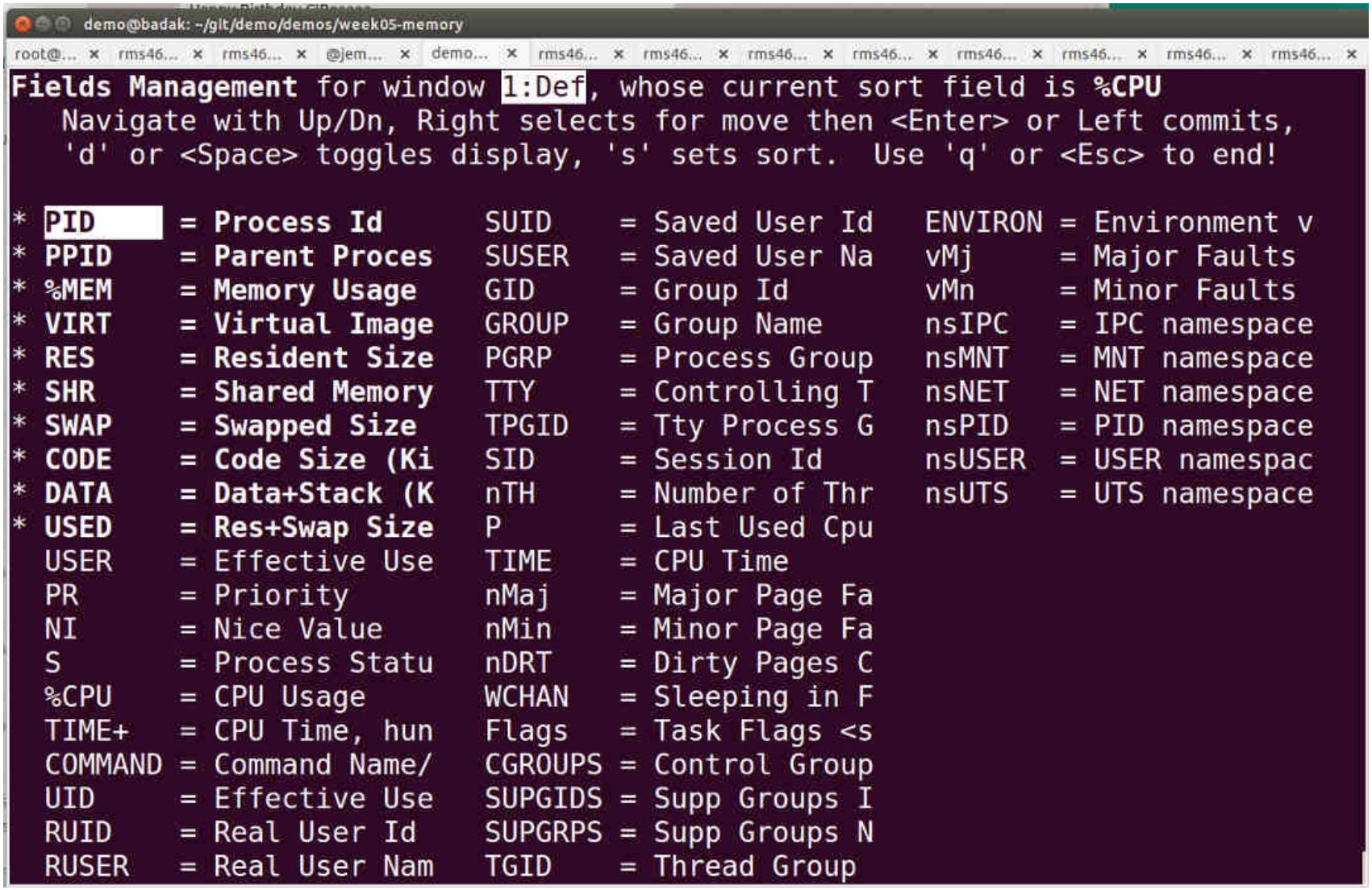

Figure: Moving Fields

TOP (5)

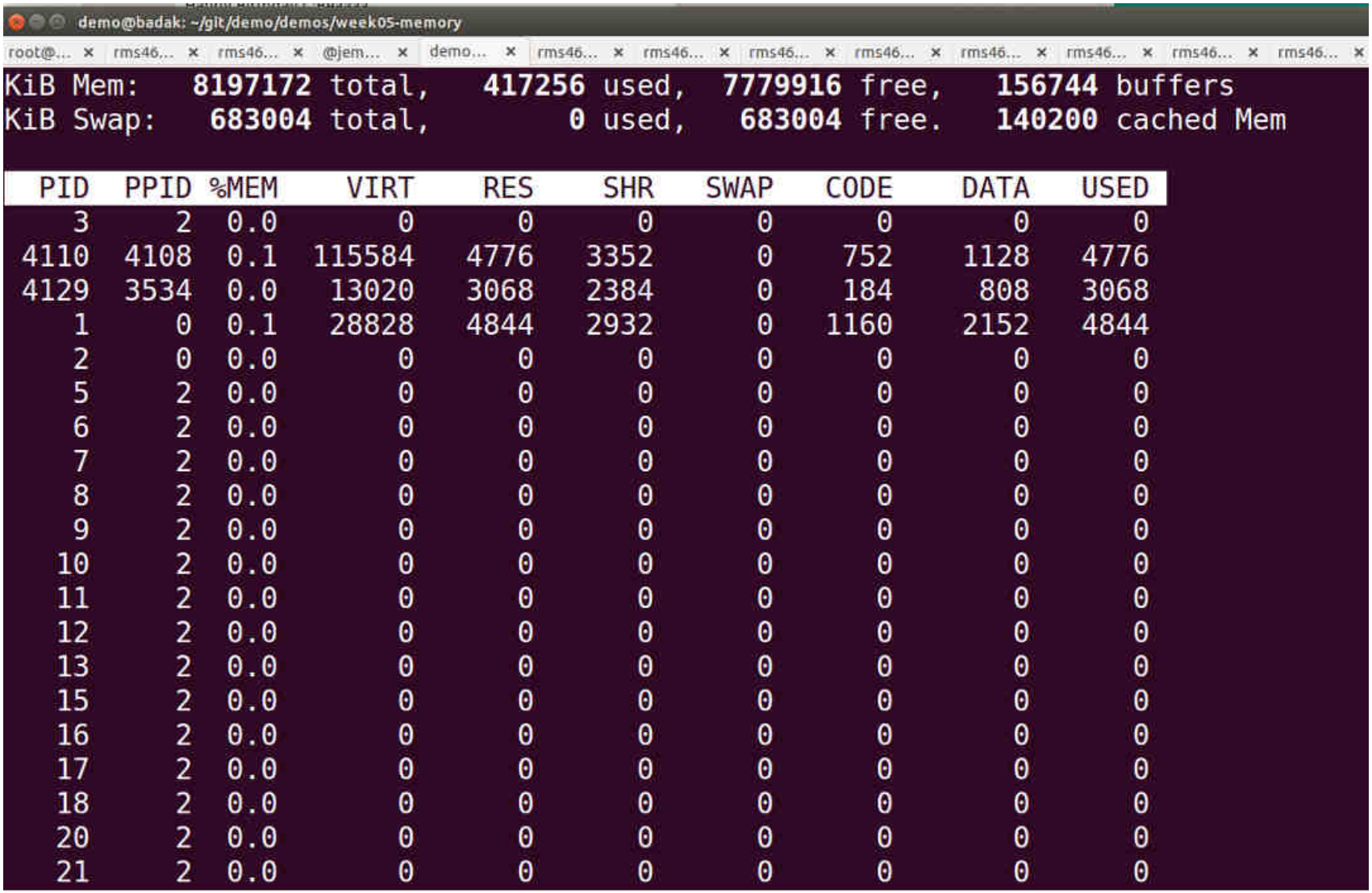

Figure: Memory Information

06-memory

```
#define MSIZE1 0x10000
#define MSIZE2 0x20000
#define MSIZE3 0x50000
#define MSIZE4 0x100000
#define MSIZE5 0x1000000
#define MSIZE6 0x10000000
#define LINE   75
#define MAXSTR 80
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

void printLine(int line) {
   while(line-- > 0) putchar('x');
   putchar('\n');
   fflush(NULL);
}
void main (void) {
   char  strSYS2[MAXSTR], strSYS1[MAXSTR];
   char* chrStr = &strSYS1[0];
   int   ii, myPID  = getpid();
   sprintf(strSYS2, "top -b -n 1 -p%d | tail -5", myPID);
   sprintf(strSYS1, "top -b -n 1 -p%d | tail -1", myPID);
   printf("MSIZE1 (10k) MSIZE2 (20k) MSIZE3 (50k) MSIZE4 (100k) MSIZE5 (1M) MSIZE6 (10M) MSIZE1\n");
   printLine(LINE);
```

06-memory (2)

```
   // sprintf(strSYS2, "top -b -n 1 -p%d | tail -5", myPID);
   // sprintf(strSYS1, "top -b -n 1 -p%d | tail -1", myPID);
   system(strSYS2);            /* (1) */
   chrStr = malloc(MSIZE1);
   system(strSYS1);            /* (2) */
   free(chrStr);
   chrStr = malloc(MSIZE2);
   system(strSYS1);            /* (3) */
   free(chrStr);
   chrStr = malloc(MSIZE3);
   system(strSYS1);            /* (4) */
   free(chrStr);
   chrStr = malloc(MSIZE4);
   system(strSYS1);            /* (5) */
   free(chrStr);
   chrStr = malloc(MSIZE5);
   for (ii = 0; ii < MSIZE5; ii++) {
      chrStr[ii]='a';
   }
   system(strSYS1);            /* (6) */
   free(chrStr);
   chrStr = malloc(MSIZE6);
   system(strSYS1);            /* (7) */
   free(chrStr);
   chrStr = malloc(MSIZE1);
   for (ii = 0; ii < MSIZE1; ii++) {
      chrStr[ii]='a';
   }
   system(strSYS1);            /* (8) */
   free(chrStr);
   printLine(LINE);
}
```

06-memory (2)

```
>>>>> $ 06-memory
(1) START          (2) MSIZE1=10k      (3) MSIZE2=20k  (4) MSIZE3=50k
(5) MSIZE4=100k    (6) MSIZE5=1M (F)   (7) MSIZE6=10M  (8) MSIZE1=10k (F)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
KiB Mem:   8197160 total,   341564 used,  7855596 free,    50776 buffers
KiB Swap:   683004 total,        0 used,   683004 free.   195692 cached

  PID  PPID %MEM    VIRT    RES    SHR   SWAP   CODE    DATA   USED
 1567  1185  0.0    4172    688    612      0      4     320    688 (1)
 1567  1185  0.0    4172    688    612      0      4     320    688 (2)
 1567  1185  0.0    4172    688    612      0      4     320    688 (3)
 1567  1185  0.0    4496    688    612      0      4     644    688 (4)
 1567  1185  0.0    5200   1212   1116      0      4    1348   1212 (5)
 1567  1185  0.2   20560  17576   1116      0      4   16708  17576 (6)
 1567  1185  0.0  266320   1212   1116      0      4  262468   1212 (7)
 1567  1185  0.0    4172   1212   1116      0      4     320   1212 (8)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

The End

- ☐ This is the end of the presentation.
- ☒ This is the end of the presentation.
- This is the end of the presentation.