VE482
Introduction to Operating Systems

HOMEORK 2

October 10, 2021

Yihua Liu 518021910998

Ex. 1 — Multiprogramming

A few years ago when computers featured less RAM it is was common to increase it in order to enhance CPU performance. In order to better understand the link between the two we now create a simple model for multiprogramming. We assume all the processes to be similar and spending the same fraction p of their time waiting for Input/Output (I/O) to complete.

1. What is the probability for n processes to be waiting at the same time, then express the CPU utilisation as a function of n?
   The probability for n processes to be waiting at the same time is p^n.
   The CPU utilisation as a function of n is $1 - p^n$.

2. Sketch the curve representing the CPU utilisation as a function of the number of processes for the following values of p: 25%, 60% and 90%.

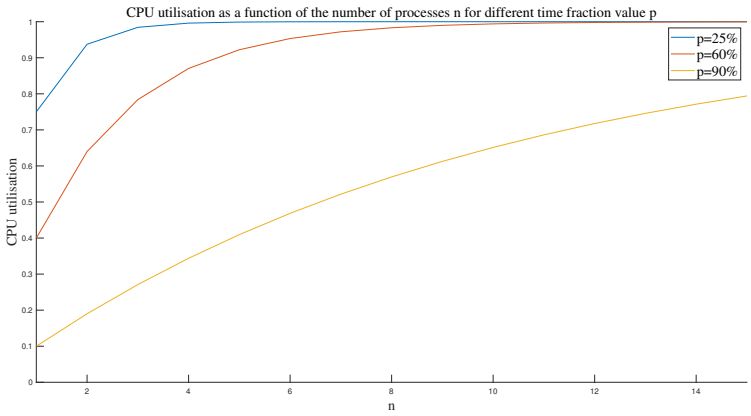

Figure 1. CPU utilisation as a function of the number of processes n for different time fraction value p.

3. A certain old computer has 256 MB of RAM, once loaded a light operating

system uses 96 MB of RAM. Several programs are launched each of them using 48 MB.

(a) How many processes can be store simultaneously in memory?

$$(256 - 48)/48 = 3.$$

3 processes can be store simultaneously in memory.

(b) Assuming an average of 90% I/O waiting time what is the CPU utilisation?

$$1 - 0.9^3 = 0.271 = 27.1\%.$$

The CPU utilisation is 27.1%.

(c) What is the effect of adding 256 MB, 512 MB and 1024 MB of RAM. Argue on which amount would be the most beneficial and would be worth the investment.
We assume for the three RAMs, the price per MB is the same. To choose the most beneficial one, we need to consider costs, which can be measured in CPU utilisation per 256 MB that are added. Adding 256 MB: Number of simultaneous processes is $n = \lfloor (512 - 96)/48 \rfloor = 8$, CPU utilisation is $1 - 0.9^8 \approx 56.95\%$, efficiency improved by 56.95 - 27.1% = 29.85% per 256 MB that are added.
Adding 512 MB: Number of simultaneous processes is $n = (768 - 96)/48 = 14$, CPU utilisation is $1 - 0.9^{14} \approx 77.12\%$, efficiency improved by (77.12% - 27.1%) / 2 = 25.01% per 256 MB that are added.
Adding 1024 MB: Number of simultaneous processes is $n = \lfloor (1280 - 96)/48 \rfloor = 24$, CPU utilisation is $1 - 0.9^{24} \approx 92.02\%$, efficiency improved by (92.02% - 27.1%) / 4 = 16.23% per 256 MB that are added.
Therefore, adding 256 MB would be most beneficial and would be worth the investment.

Ex. 2 — Keymap in Minix 3

Map the Shit+F7 key to displaying how many processes are currently running. This can be achieved following the steps:

1. Find all the files that need to be modified;
   `/usr/src/servers/is/dmp.c`, `/usr/src/servers/is/proto.h`, `/usr/src/servers/is/dmp_kernel.c`
2. Locate the part of the code that should be updated;
   At Line 33 of `/usr/src/servers/is/dmp.c`, at Line 21 of `/usr/src/servers/is/proto.h`, at the end of `/usr/src/servers/is/dmp_kernel.c`.
3. Check how to process table is stored;
4. Count the number of running processes;

   At Line 21 of `/usr/src/servers/is/proto.h`, add one line:

```
1       void procnum_dmp(void);
```

```
/* main.c */
int main(int argc, char **argv);

/* dmp.c */
void map_unmap_fkeys(int map);
int do_fkey_pressed(message *m);
void mapping_dmp(void);
void vm_dmp(void);

/* dmp_kernel.c */
void proctab_dmp(void);
void procstack_dmp(void);
void privileges_dmp(void);
void image_dmp(void);
void irqtab_dmp(void);
void kmessages_dmp(void);
void monparams_dmp(void);
void kenv_dmp(void);
void procnum_dmp(void);

/* dmp_pm.c */
void mproc_dmp(void);
/usr/src/servers/is/proto.h: 35 lines, 638 characters.
#
```

At the end of `/usr/src/servers/is/dmp_kernel.c`, add a function:

```
1       /*==========================================================
        ↪  ========================*
2        *                                    procnum_dmp
        ↪                                       *
3        *==========================================================
        ↪  ========================*/
4       void procnum_dmp()
5       {
6         register struct proc *rp;
7         int r,n=0;
8
9         /* First obtain a fresh copy of the current process
           ↪  table. */
10        if ((r = sys_getproctab(proc)) != OK) {
11          printf("IS: warning: couldn't get copy of process
             ↪  table: %d\n", r);
12          return;
13        }
14
15        for (rp=BEG_PROC_ADDR; rp<END_PROC_ADDR; rp++) {
16            if (isemptyp(rp))
17              continue;
18            n++;
19        }
20
```

```
21          printf("Number of currently running processes is
            ↪  %d\n", n);
22      }
```

```
 *                              procnum_dmp                                   *
 *============================================================================*/
void procnum_dmp()
{
  register struct proc *rp;
  int r,n=0;

  /* First obtain a fresh copy of the current process table. */
  if ((r = sys_getproctab(proc)) != OK) {
    printf("IS: warning: couldn't get copy of process table: %d\n", r);
    return;
  }

  for (rp=BEG_PROC_ADDR; rp<END_PROC_ADDR; rp++) {
    if (isemptyp(rp))
      continue;
    n++;
  }

  printf("Number of currently running processes is %d\n", n);
}
~
/usr/src/servers/is/dmp_kernel.c: 413 lines, 12636 characters.
# _
```

5. Map the newly written function to the Shift-F7 key; At Line 33 of `/usr/src/servers/is/dmp.c`, add one line:

```
1       { SF7,  procnum_dmp, "Print number of currently
        ↪  running processes" },
```

```
#include <minix/vm.h>

struct hook_entry {
        int key;
        void (*function)(void);
        char *name;
} hooks[] = {
        { F1,   proctab_dmp, "Kernel process table" },
        { F3,   image_dmp, "System image" },
        { F4,   privileges_dmp, "Process privileges" },
        { F5,   monparams_dmp, "Boot monitor parameters" },
        { F6,   irqtab_dmp, "IRQ hooks and policies" },
        { F7,   kmessages_dmp, "Kernel messages" },
        { F8,   vm_dmp, "VM status and process maps" },
        { F10,  kenv_dmp, "Kernel parameters" },
        { SF1,  mproc_dmp, "Process manager process table" },
        { SF2,  sigaction_dmp, "Signals" },
        { SF3,  fproc_dmp, "Filesystem process table" },
        { SF4,  dtab_dmp, "Device/Driver mapping" },
        { SF5,  mapping_dmp, "Print key mappings" },
        { SF6,  rproc_dmp, "Reincarnation server process table" },
        { SF7,  procnum_dmp, "Print number of currently running processes" },
        { SF8,  data_store_dmp, "Data store contents" },
/usr/src/servers/is/dmp.c: 127 lines, 4259 characters.
#
```

6. Recompile the kernel and test.

```
        cd /usr/src
        make build && reboot  # remember to reboot
```

```
kernel: selecting intel sysenter ipc style
Started VFS: 8 worker thread(s)
Root device name is /dev/c0d0p0s0
/dev/c0d0p0s0: clean
/dev/c0d0p0s0 is mounted on /
none is mounted on /proc
Sun Oct 10 16:38:23 GMT 2021
/dev/c0d0p0s2: clean
/dev/c0d0p0s1: clean
size on /dev/imgrd set to 0kB
Multiuser startup in progress ...
Starting hotplugging infrastructure... done.
Starting services: random lance inet printer ipc.
Starting daemons: update cron syslogd.
Starting networking: dhcpd nonamed.
Local packages (start): rsyncd rsyncd: WARNING: $rsyncd is not set properly - se
e rc.conf(5).
$rsyncd is not enabled - see rc.conf(5).
Use the following if you wish to perform the operation:
  rsyncd onestart
sshd Starting sshd.
 done.
 Minix  Release 3 Version 2.1  (console)

192.168.36.136 login: _
```

```
Password:
Last login: Sun Oct 10 15:19:35 2021 on console
Copyright (c) 2012, Vrije Universiteit, Amsterdam, The Netherlands


To install additional packages, run 'pkgin'.

To install packages from the online package repository, if you have a
working network connection from MINIX: first do a 'pkgin update' to
update the list of available packages, and then do a 'pkgin' to get a
list of commands.  For example, 'pkgin install vim' installs the 'vim'
package, and 'pkgin available' will list all available packages.

To install packages from the installation CD: same, but use pkgin_cd.
To switch to the online repository, do 'pkgin update' again. To install
all packages, do pkgin_all.

MINIX 3 supports multiple virtual terminals. Just use ALT+F1, F2, F3
and F4 to navigate among them.

For more information on how to use MINIX 3, see the wiki:
http://wiki.minix3.org.

# Number of currently running processes is 42
_
```