UM–SJTU Joint Institute

Introduction to Operating Systems

(VE482)

Homework 2

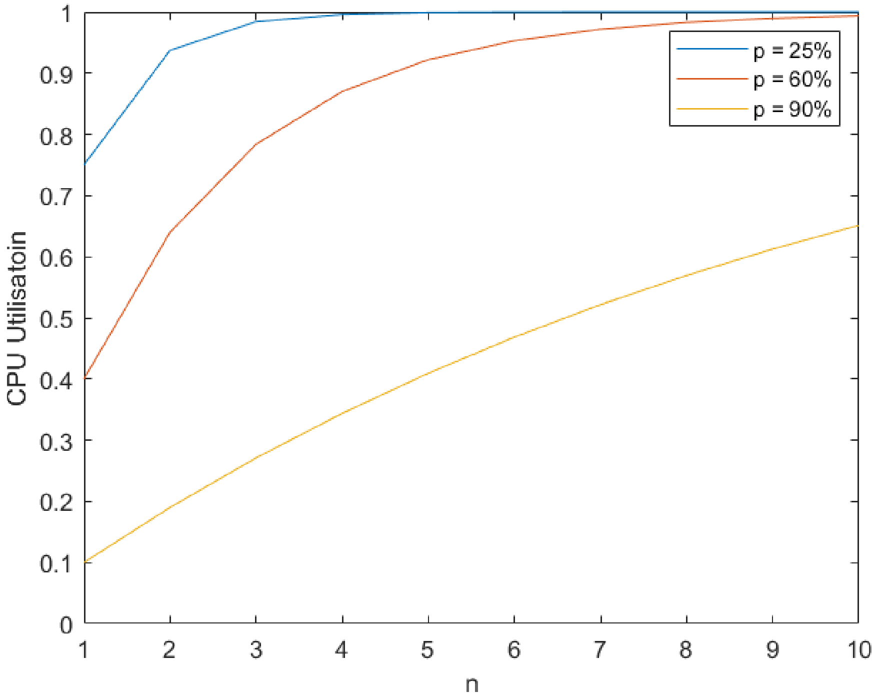
Name: Yu Xiao ID: 518021910696

Date: Oct 9, 2021

Ex.1 – *MultiProgramming*

1. The probability for n processes to be waiting at the same time is p^n.
   The CPU utilisation is $1 - p^n$.

2. The curve representing CPU utilisation with different p is shown below.



3. a) After loaded the light operating system, we know that

$$\lfloor (256 - 96) \div 48 \rfloor = 3$$

   Therefore, 3 processes can be stored simultaneously in memory.

   b) From previous question, we know that the CPU utilisation can be calculated by

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

   Therefore, the CPU utilisation in this case is 27.1%.

   c) If 256 MB is added, $\lfloor (512 - 96) \div 48 \rfloor = 8$ processes can be stored simultaneously in memory, the CPU utilisation will be $1 - 0.9^8 \approx 56.95\%$, which increases by $(56.95\% \div 2) - 27.1\% = 29.85\%$ per 256 MB.
   If 512 MB is added, $\lfloor (768 - 96) \div 48 \rfloor = 14$ processes can be stored simultaneously in memory, the CPU utilisation will be $1 - 0.9^14 \approx 77.12\%$, which decreases by $|(77.12\% \div 3) - 27.1\%| = 1.39\%$ per 256 MB.
   If 1024 MB is added, $\lfloor (1280 - 96) \div 48 \rfloor = 24$ processes can be stored simultaneously in memory, the CPU utilisation will be $1 - 0.9^24 \approx 92.02\%$, which decreases by $|(92.02\% \div 5) - 27.1\%| = 8.7\%$.
   As a result, adding 256 MB will be the most beneficial and worth the investment.

Ex.2 – *Keymap in Minix 3*
There are three files that should be edited.
In `/usr/src/servers/is/dump.c`, add

```
{ SF7,  pscount_dmp, "Display how many processes are running" },
```

The second one is /usr/src/servers/is/dmp_kernel.c, add the pscount_dmp function into it:

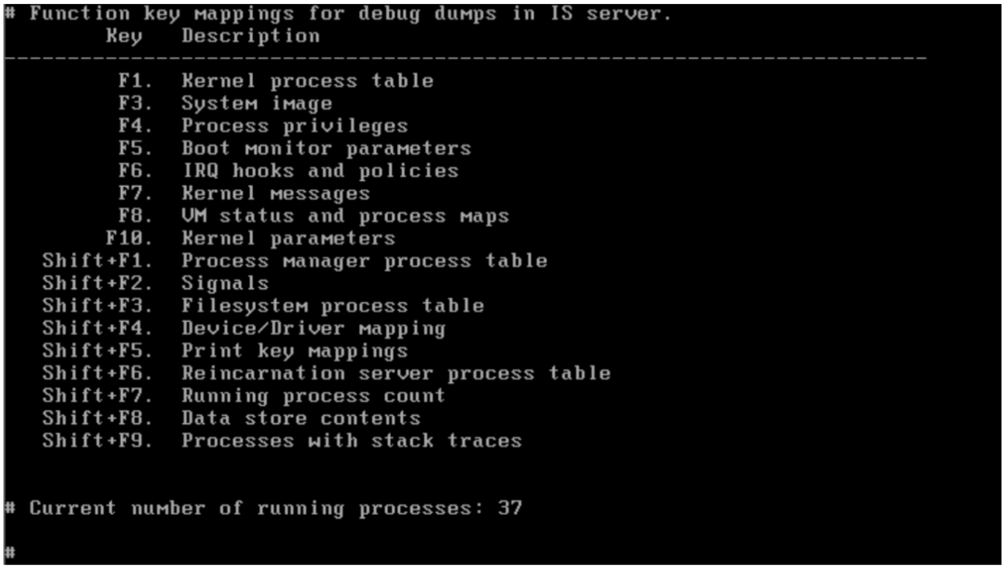
```
#include "../pm/mproc/h"

/*===========================================================================*
 *                               pscount_dmp                                 *
 *===========================================================================*/
void pscount_dmp()
{
 struct mproc *mp;
 int i,n = 0;
 if (getsysinfo(PM_PROC_NR, SI_PROC_TAB, mproc, sizeof(mproc)) != OK) {
    printf("Error obtaining table from PM. Perhaps recompile IS?\n");
    return;
 }
 for (i = 0; i < NR_PROCS; i++) {
    mp = &mproc[i];
    if (mp->mp_pid == 0 && i != PM_PROC_NR) continue;
    n++;
 }
 printf("Current number of running processes: %d\n", n);
}
```

The third one is /usr/src/servers/is/proto.h, add the declaration of the function into it:

```
/* dmp_kernel.c */
void pscount_dmp (void);
```

Then, build the changes and reboot the system

```
cd /usr/src
make build
reboot
```



```
# Function key mappings for debug dumps in IS server.
        Key   Description
-------------------------------------------------------------------------
         F1.  Kernel process table
         F3.  System image
         F4.  Process privileges
         F5.  Boot monitor parameters
         F6.  IRQ hooks and policies
         F7.  Kernel messages
         F8.  VM status and process maps
        F10.  Kernel parameters
   Shift+F1.  Process manager process table
   Shift+F2.  Signals
   Shift+F3.  Filesystem process table
   Shift+F4.  Device/Driver mapping
   Shift+F5.  Print key mappings
   Shift+F6.  Reincarnation server process table
   Shift+F7.  Running process count
   Shift+F8.  Data store contents
   Shift+F9.  Processes with stack traces


# Current number of running processes: 37

#
```