

---

UM-SJTU JOINT INSTITUTE  
INTRODUCTION TO OPERATING SYSTEMS  
(VE482)

---

HOMEWORK 2

Name: Ji Xingyou      ID: 515370910197  
Date: 28 September 2017

### Ex.1 – Multiprogramming

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

**probability:**  $p^n$

**CPU utilisation:**  $1 - p^n$

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%.

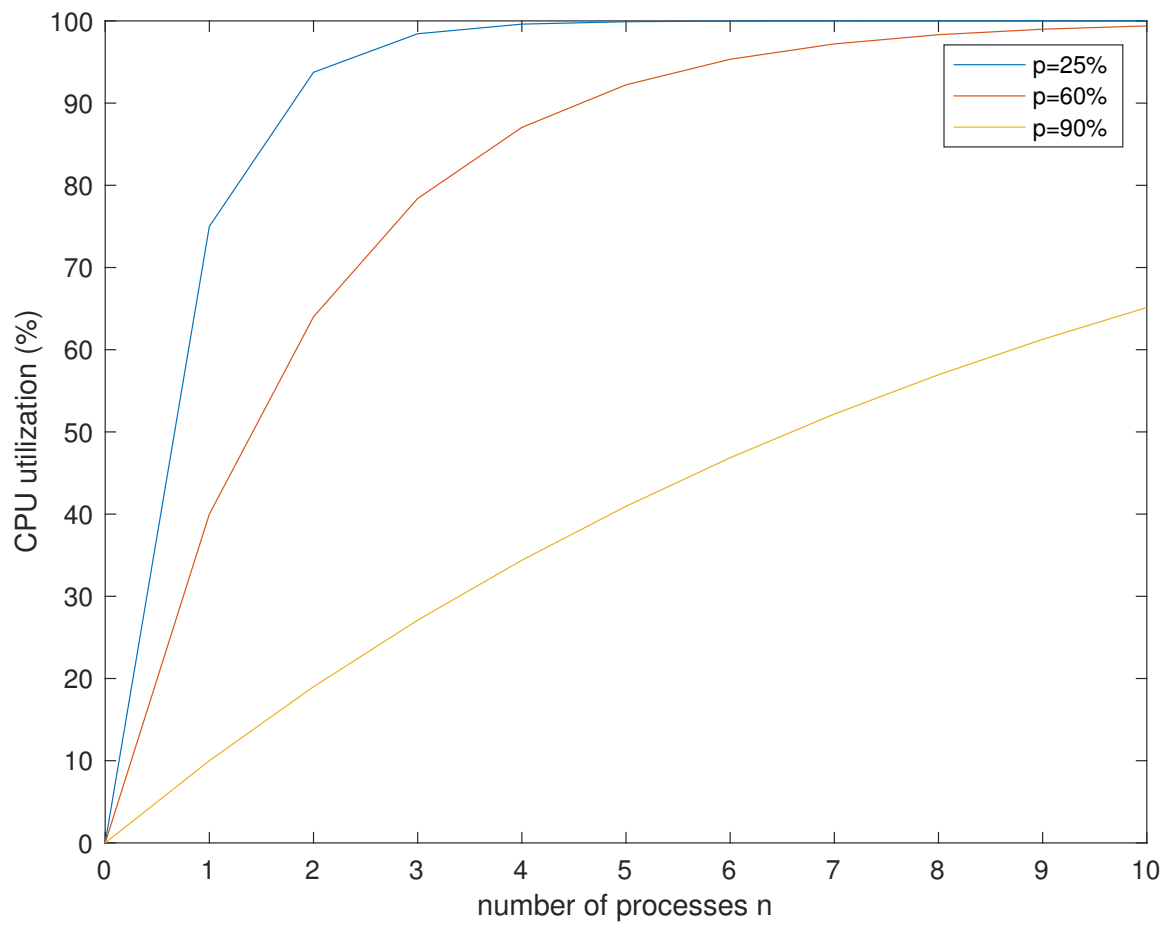


Figure 1: CPU utilization vs n

Below is the Matlab code.

```
1 clear all;clc;
2 syms n CPU_utilization1;
3 n=0:1:10;
4 p1=0.25;
5 p2=0.6;
6 p3=0.9;
7 CPU_utilization1=(1-p1.^n)*100;
8 plot(n, CPU_utilization1);
9 hold on;
10 CPU_utilization2=(1-p2.^n)*100;
11 plot(n, CPU_utilization2);
12 hold on;
13 CPU_utilization3=(1-p3.^n)*100;
14 plot(n, CPU_utilization3);
15 xlabel('number of processes n');
16 ylabel('CPU utilization (%)');
17 legend('p=25%', 'p=60%', 'p=90%');
```

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.

- How many processes can be store simultaneously in memory?

$$n = \left\lfloor \frac{256-96}{48} \right\rfloor = \left\lfloor \frac{160}{48} \right\rfloor = 3$$

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

$$CPU_{utilization} = 1 - p^n = 1 - 90\%^3 = 27.1\%$$

- 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.

**256 MB:**

$$n = 3 + \frac{256}{48} = 3 + 5 = 8$$

$$CPUUtilization = 1 - p^n = 1 - 90\%^8 = 56.95\%$$

$$\Delta = 56.95 - 27.1 = 29.85\%$$

**512 MB:**

$$n = 3 + \frac{512}{48} = 3 + 10 = 13$$

$$CPUUtilization = 1 - p^n = 1 - 90\%^{13} = 74.58\%$$

$$\Delta = 74.58 - 56.95 = 17.63\%$$

**1024 MB:**

$$n = 3 + \frac{1024}{48} = 3 + 21 = 24$$

$$CPUUtilization = 1 - p^n = 1 - 90\%^{24} = 92.02\%$$

$$\Delta = 92.02 - 74.58 = 17.44\%$$

Therefore, adding 256 MB is most beneficial.

**Ex.2 – Keymap in Minix 3**

First, in the dmp.c file, add SF7.

```

1  struct hook_entry {
2      int key;
3      void (*function)(void);
4      char *name;
5  } hooks[] = {
6      { F1,    proctab_dmp, "Kernel process table" },
7      { F3,    image_dmp,  "System image" },
8      { F4,    privileges_dmp, "Process privileges" },
9      { F5,    monparams_dmp, "Boot monitor parameters" },
10     { F6,    irqtab_dmp,  "IRQ hooks and policies" },
11     { F7,    kmessages_dmp, "Kernel messages" },
12     { F8,    vm_dmp,      "VM status and process maps" },

```

```

13     { F10,   kenv_dmp,  "Kernel parameters" },
14     { SF1,   mproc_dmp, "Process manager process table" },
15     { SF2,   sigaction_dmp, "Signals" },
16     { SF3,   fproc_dmp, "Filesystem process table" },
17     { SF4,   dtab_dmp,  "Device/Driver mapping" },
18     { SF5,   mapping_dmp, "Print key mappings" },
19     { SF6,   rproc_dmp, "Reincarnation server process table" },
20     { SF7,   proc_num_dmp, "Currently running processes number" }
21     { SF8,   data_store_dmp, "Data store contents" },
22     { SF9,   proystack_dmp, "Processes with stack traces!" },
23 };

```

Then, in the proto.h, add the declaration of the proc\_num\_dmp function.

```

1  /* dmp_kernel.c */
2  void proc_num_dmp(void);
3  void proctab_dmp(void);
4  void proystack_dmp(void);
5  void privileges_dmp(void);
6  void image_dmp(void);
7  void irqtab_dmp(void);
8  void kmessages_dmp(void);
9  void monparams_dmp(void);
10 void kenv_dmp(void);

```

Finally, in the dmp\_kernel.c, add the implementation of the proc\_num\_dmp function.

```

1  void proc_num_dmp(void)
2  {
3      register struct proc *rp;

```

```

4      int r;
5      if ((r=sys_getproctab(proc))!=OK)
6      {
7          printf("IS: warning: couldn't get copy of process table: %d\n", r);
8          return;
9      }
10
11     int num=0;
12     for (rp=BEG.PROC_ADDR; rp<END.PROC_ADDR; rp++)
13     {
14         if (isempty(rp))
15         {
16             continue;
17         }
18         num++;
19     }
20     printf("Currently running processes number is: %d\n", num);
21 }

```