VE482 - Homework 2

Introduction to Operating Systems

Kexuan Huang 518370910126

October 16, 2021

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?

Solution:

1) Since 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, for n process, the probability of waiting at the same time is p^n.

2) As a result, the CPU utilisation is given by $1 - p^n$.

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%.

Solution:

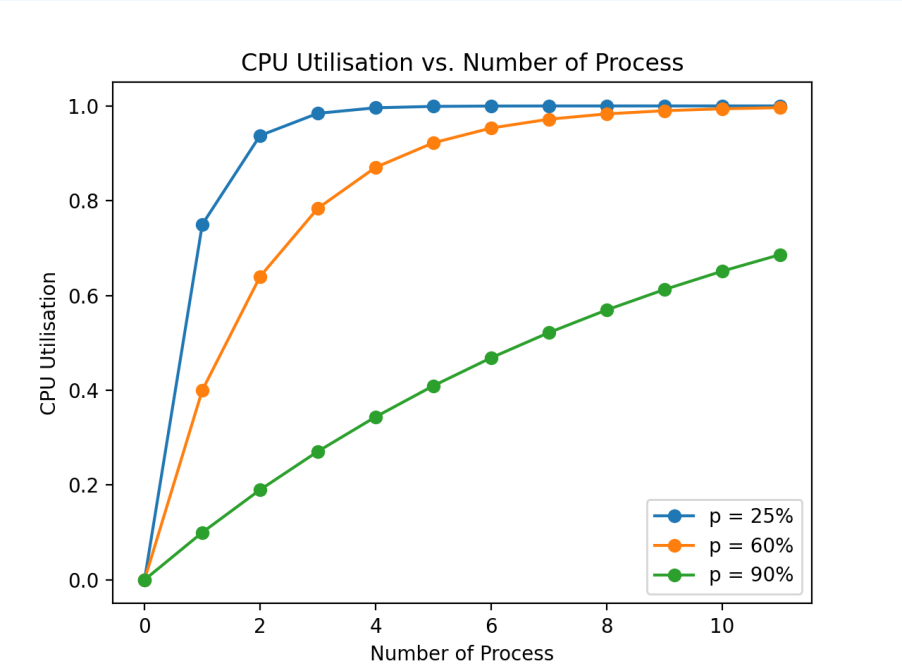

Figure 1: the CPU utilisation as a function of the number of processes for the following values of p: 25%, 60% and 90%

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.

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

Solution:
The maximum process number can be stored simultaneously in memory is calculated as

$$\text{The maximum process number} := \left\lfloor \frac{256-96}{48} \right\rfloor = 3.$$

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

Solution:
With the equation deduced in 1.2, the CPU utilisation can be calculated as

$$\text{The CPU utilisation} := 1 - p^n = 1 - 90\%^3 = 27.1\%.$$

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.

Solution:

Assume: all the processes have an average of 90% I/O waiting time.

1) For adding 256 MB, the maximum process number is $\left\lfloor \frac{256+256-96}{48} \right\rfloor = 8$.
   The CPU utilisation is $1 - 90\%^8 = 56.95\%$. The gain per 256 MB is 29.85%.
2) For adding 512 MB, the maximum process number is $\left\lfloor \frac{256+512-96}{48} \right\rfloor = 14$.
   The CPU utilisation is $1 - 90\%^{14} = 77.12\%$. The gain per 256 MB is 25.01%.
3) For adding 1024 MB, the maximum process number is $\left\lfloor \frac{256+1024-96}{48} \right\rfloor = 24$.
   The CPU utilisation is $1 - 90\%^{24} = 92.02\%$. The gain per 256 MB is 16.23%.

As a result, adding 256 MB would be the most beneficial and worth the investment.

Ex.2 - Keymap in Minix 3

Map the Shit+F7 key to displaying how many processes are currently running

Solution:

1) Declare process-counting function in `/usr/src/servers/is/proto.h`

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

3) Implement process-counting function in `/usr/src/servers/is/dmp_kernel.c`

```
void proccnt_dmp() {
  int r;
  if((r = sys_getproctab(proc)) != OK) {
    printf("IS: warning: couldn't get copy of process table: %d\n"
        , r);
  }
  int proc_cnt = 0;
  register struct proc *rp; // pointer to task's proc entry
  for(rp = BEG_PROC_ADDR; rp < END_PROC_ADDR; rp++) {
    if(isemptyp(rp)) {
      continue;
    }
    proc_cnt++; // count the number of running processes
  }
  printf("The number of currently running process is %d\n",
     proc_cnt); // print the number of running processes
}
```

3) Map the newly written function to the Shift-F7 key in `/usr/src/servers/is/dmp.c`

```
struct hook_entry {
  int key;
  void (*function)(void);
  char *name;
} hooks[] = {
  ...
  { SF7, proccnt_dmp, "Count currently running processes" },
}
```

4) Recompile and reboot Minix

```
#!/bin/sh
cd /usr/src
make build  // recompilte the kernel
reboot      // reboot Minix
```

5) Test the Shift-F7 key mapping

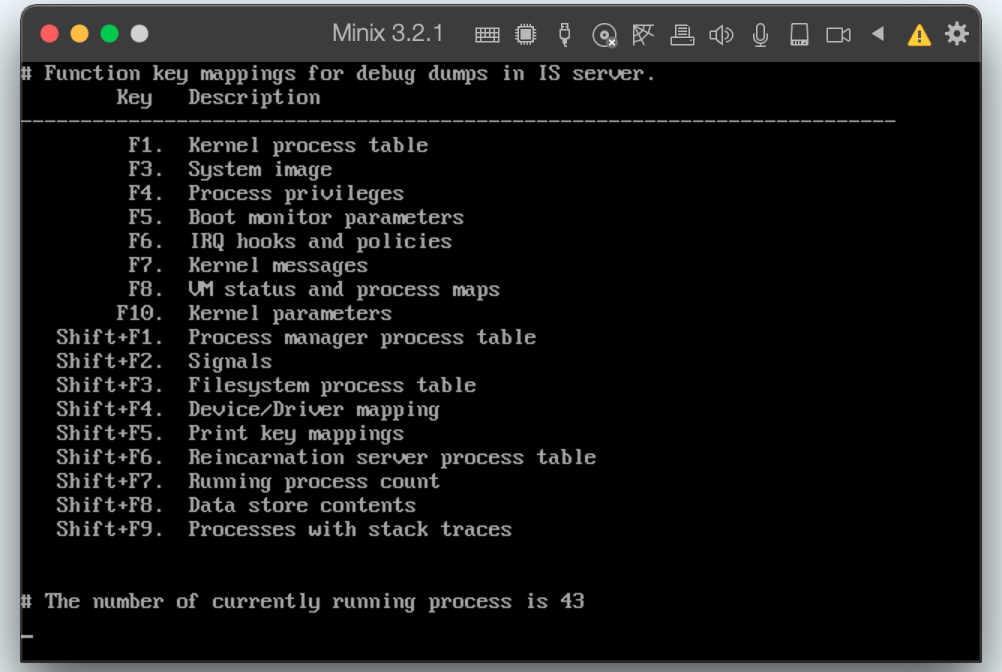

Figure 2: Function key mappings and Shift-F7 key mapping test