

VE482 Homework 2

Lan Wang 518370910084

Ex 1

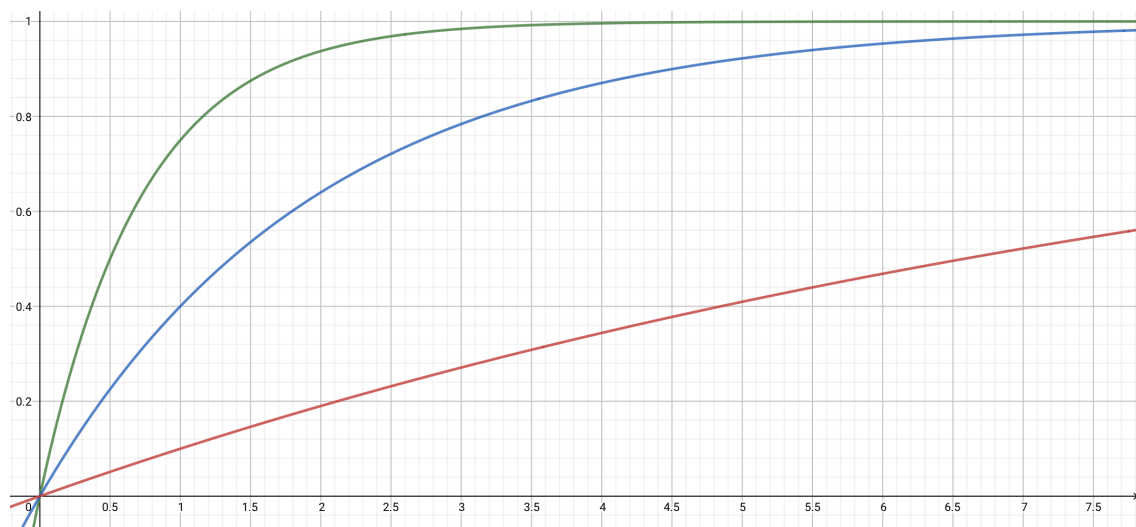
A few years ago when computers featured less RAM it was common to increase it in order to enhance CPU performance. In order to better understand the link between the two we now create a simple model for multiprogramming. We assume all the processes to be similar and spending the same fraction p of their 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 is: p^n

CPU utilisation 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%.



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?

$$(256 - 96) \div 48 = 3.33 \Rightarrow 3 \text{ processes}$$

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

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

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.

o add 256 MB:

$$(512 - 96) \div 48 = 8.67 \Rightarrow 8 \text{ processes}$$

$$\text{CPU utilisation: } 1 - 0.9^8 = 0.570 = 57.0\%$$

Improvement: 29.9% per 256 MB

o add 512 MB:

$$(768 - 96) \div 48 = 14 \Rightarrow 14 \text{ processes}$$

CPU utilisation: $1 - 0.9^{14} = 0.771 = 77.1\%$

Improvement: 25.0% per 256 MB

- add 1024 MB:

$(1280 - 96) \div 48 = 24.67 \Rightarrow 24$ processes

CPU utilisation: $1 - 0.9^3 = 0.920 = 92.0\%$

Improvement: 16.23% per 256 MB

So adding 256 MB would be the most beneficial and would be worth the investment.

Ex 2

1. Modify the `dmp.c` file (`/usr/src/servers/is/dmp.c`). Add the hook entry for Shift+F7.

```
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, "Display number of running processes"}, // added to
the file
    { SF8,  data_store_dmp, "Data store contents" },
    { SF9,  procstack_dmp, "Processes with stack traces" },
};
```

2. Modify `dmp_kernel.c` file (`/usr/src/servers/is/dmp_kernel.c`). Add the following implementation of `procnum_dmp()`.

```
/*=====*
*                                     procnum_dmp                                     *
*=====*/
void procnum_dmp() {
    register struct proc *rp;
    static struct proc *oldrp = BEG_PROC_ADDR;
    int r;

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

```

    }
    int num = 0;
    for (rp = oldrp; rp < END_PROC_ADDR; rp++) {
        oldrp = BEG_PROC_ADDR;
        if (isempty(rp)) continue;
        num++;
    }
    printf("Number of running processes: %d\n",num);
}

```

3. Modify the `proto.h` file (`/usr/src/servers/is/proto.h`). Add `void procnum_dmp(void);` to it.

```

/* 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); // Added

```

4. Rebuild

```

cd /usr/src/releasetools
make hdboot

```

5. Press Shift+F7, the result is as following:

```

# Number of running processes: 42

```