UM-SJTU JOINT INSTITUTE
VE482 Introduction to Operating Systems

Homework 2

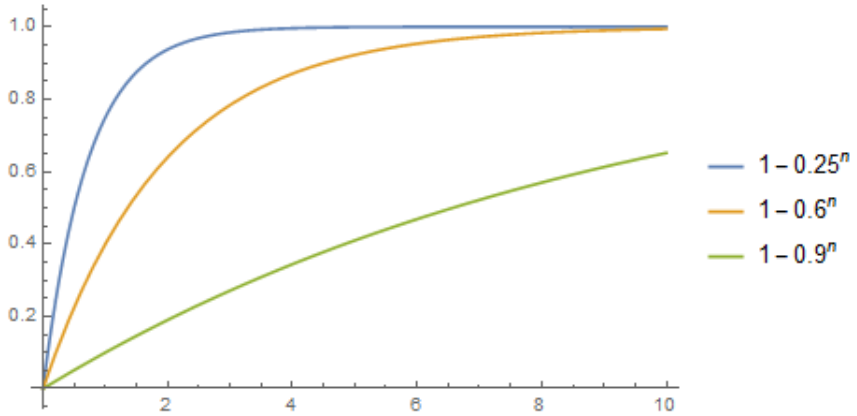
Li Yong 517370910222

October 3, 2020

Ex.1 Multiprogramming

1. P[process to be waiting] $= p$
   $P[n$ processes to be waiting$] = p^n$
   utilisation$(n) = 1 - p^n$

2. Plot by *Mathematica*



3. a)
$$(256 - 96)/48 = 3.33$$
Hence 3 processes can be store simultaneously in memory.

b)
$$1 - 0.9^3 = 0.271$$

c) Adding 256MB:
$$(256 + 256 - 96)/48 = 8.67$$
$$1 - 0.9^8 = 0.5695$$
$$0.2848/256MB$$
Adding 512MB:
$$(256 + 512 - 96)/48 = 14$$
$$1 - 0.9^{14} = 0.7712$$
$$0.2571/256MB$$
Adding 1024MB:
$$(256 + 1024 - 96)/48 = 24.67$$
$$1 - 0.9^{24} = 0.9202$$
$$0.1840/256MB$$
Hence, adding 256MB is the most beneficial and would be worth the investment.

Keymap in Minix 3

1. Add the keymap in `/usr/src/servers/is/dmp.c`

```
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" },
    //Shift+F7 keymap
    { SF7, procnum_dmp, "Print the number of currently running process"},
    { SF8, data_store_dmp, "Data store contents" },
    { SF9, procstack_dmp, "Processes with stack traces" },
};
```

2. Declare the function `procnum_dmp()` in `/usr/src/servers/is/proto.h`

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

3. Implement `procnum_dmp()` in `/usr/src/servers/is/dmp_kernel.c`

```
/*===========================================================*
*                        procnum_dmp                        *
*===========================================================*/
void procnum_dmp()
{
   struct proc *runningProc;
   int procNum = 0;
   int r;
   if ((r = sys_getproctab(proc)) != OK)
   {
```

```
11          printf("IS: warning: couldn't get copy of
                process table: %d\n", r);
12          return;
13      }
14      for (runningProc = BEG_PROC_ADDR; runningProc <
            END_PROC_ADDR; runningProc++)
15      {
16          if (!isemptyp(runningProc))
17          {
18              procNum++;
19          }
20      }
21      printf("The number of currently running process: %
            d\n", procNum);
22  }
```

4. Rebuild

```
1  cd /usr/src/releasetools
2  make hdboot
```

5. Demo

```
# The number of currently running process: 41
```