HOME ASSIGNMENT – 2
ID : 12341550

KERNEL SIDE

STEP 1

FILE : procinfo.h

```
#ifndef _PROCINFO_H_
#define _PROCINFO_H_
typedef unsigned long long uint64;

struct proc_info {
int pid;
char name[16];
char state[16];
uint64 sz;
};

#endif
```

STEP 2

FILE : proc.c

```
#include “procinfo.h”
...
int get_proc_info(int pid, struct proc_info *info)
{
struct proc *p;
int found = 0;

// Acquire the global process table lock to safely iterate.
acquire(&ptable.lock);

for (p = ptable.proc; p < &ptable.proc[NPROC]; p++) {
```

```
    if (p->pid == pid) {
    found = 1;

    // Copy the information.
    info->pid = p->pid;
    info->sz = p->sz;
    safestrcpy(info->name, p->name, sizeof(info->name));

    // Convert the enum state to a string.
    char *st;
    switch (p->state) {
    case UNUSED:   st = "UNUSED";   break;
    case EMBRYO:   st = "EMBRYO";   break;
    case SLEEPING: st = "SLEEPING"; break;
    case RUNNABLE: st = "RUNNABLE"; break;
    case RUNNING:  st = "RUNNING";  break;
    case ZOMBIE:   st = "ZOMBIE";   break;
    default:       st = "???";      break;
    }
    safestrcpy(info->state, st, sizeof(info->state));

    break; // Exit the loop once the process is found.
   }
  }

    // Release the lock after we are done with the process table.
    release(&ptable.lock);

     return found ? 0 : -1; // Return 0 on success, -1 on failure.
}
```

STEP 3

FILE : syscall.h

```
    #define SYS_get_proc_info 26
```

FILE : sysproc.c

```
#include “procinfo.h”
...
int
sys_get_proc_info(void)
{
  int pid;
  struct proc_info *info;

  // Fetch the first argument (int pid) from the user stack.
  if (argint(0, &pid) < 0)
    return -1;

  // Fetch the second argument (struct proc_info*) from the user
stack.
  // argptr also verifies that the pointer is valid.
  if (argptr(1, (char**)&info, sizeof(*info)) < 0)
    return -1;

  // Now we call the actual implementation with the validated
arguments.
  return get_proc_info(pid, info);
}
```

FILE : syscall.c

```
extern int sys_get_proc_info(void);
…
[SYS_get_proc_info] sys_get_proc_info,
```

USER PROGRAM

FILE : pinfo.c

```
#include "types.h"
#include "stat.h"
```

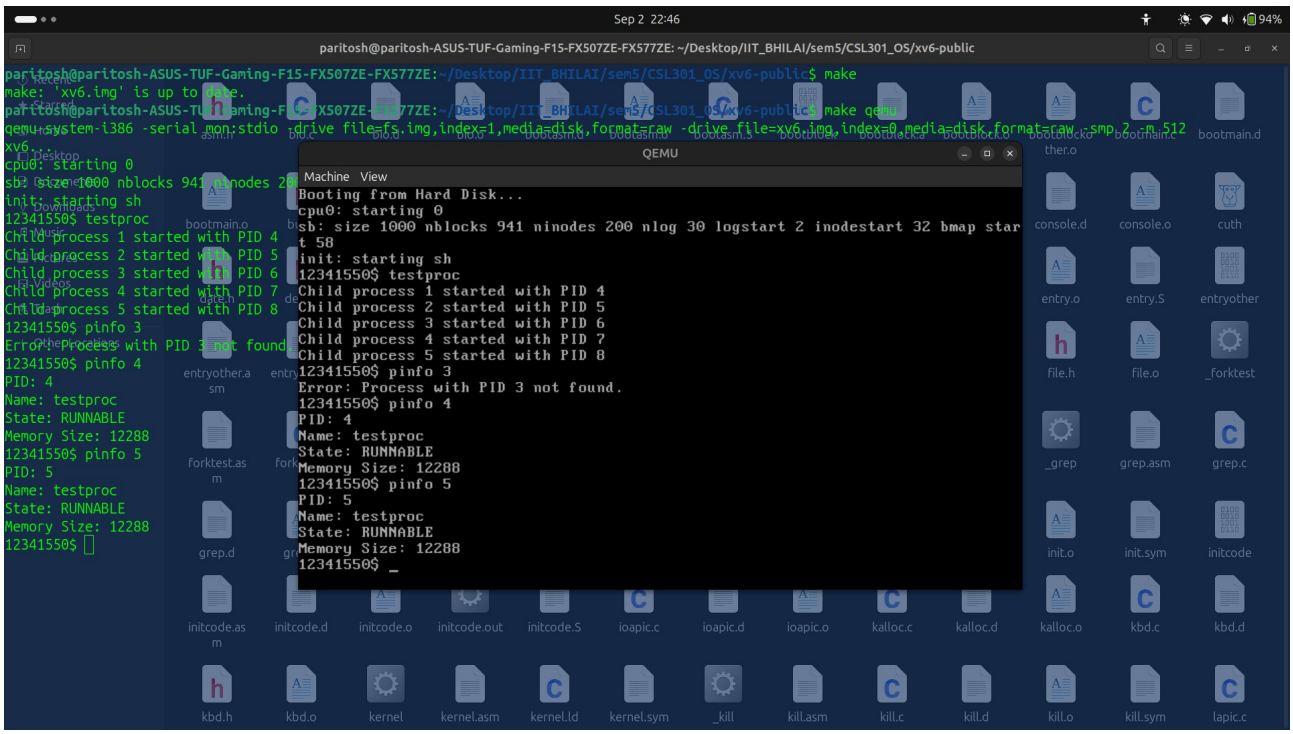
```
#include "user.h"
#include "procinfo.h"

int main(int arg, char *argv[]){

    struct proc_info info;
    int pid;
    if(arg < 2){
        printf(2,"Usage: pinfo <pi>\n");
        exit();
    }

    pid = atoi(argv[1]);

    if(get_proc_info(pid, &info) < 0){
        printf(2,"Error: Process with PID %d not found.\n",pid);
        exit();
    }else{
        printf(1, "PID: %d\n",info.pid);
        printf(1, "Name: %s\n",info.name);
        printf(1, "State: %s\n",info.state);
        printf(1, "Memory Size: %d\n",info.sz);
    }

    exit();
}
```

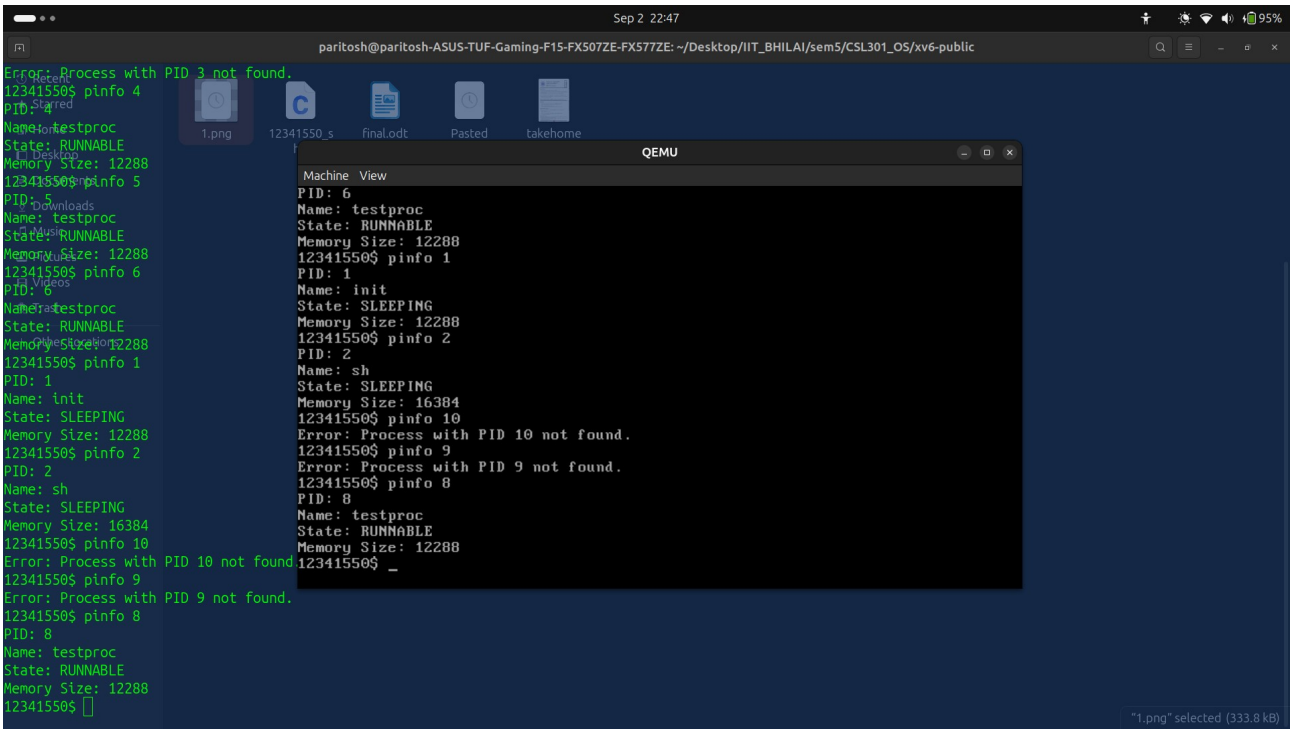
FILE : testproc.c

```
#include "user.h"
#include "types.h"
int main(void)
{
    int i;
    int num_children = 5;
    for (i = 0; i < num_children; i++)
    {
        int pid = fork();
```

```
        if (pid < 0)
        {
            printf(2,"Fork failed\n");
            exit();
        }
        if (pid == 0)
        {
            printf(1,"Child process %d started with PID %d\n", i + 1,
getpid());
            while (1);
            // never exits
        }
    }
    // Parent waits for all children to finish
    //
    /* for (i = 0; i < num_children; i++)
    {
        //
        int wpid = wait();
        //
        printf(1,"Parent: child PID %d finished\n", wpid);
        //
    } */
    exit();
}
```

OUTPUT :


Sep 2 22:46
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/xv6-public
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE:~/Desktop/IIT_BHILAI/sem5/CSL301_OS/xv6-public$ make
make: 'xv6.img' is up to date.
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE:~/Desktop/IIT_BHILAI/sem5/CSL301_OS/xv6-public$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200
init: starting sh
12341550$ testproc
Child process 1 started with PID 4
Child process 2 started with PID 5
Child process 3 started with PID 6
Child process 4 started with PID 7
Child process 5 started with PID 8
12341550$ pinfo 3
Error: Process with PID 3 not found.
12341550$ pinfo 4
PID: 4
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 5
PID: 5
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$
QEMU
Machine View
Booting from Hard Disk...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
12341550$ testproc
Child process 1 started with PID 4
Child process 2 started with PID 5
Child process 3 started with PID 6
Child process 4 started with PID 7
Child process 5 started with PID 8
12341550$ pinfo 3
Error: Process with PID 3 not found.
12341550$ pinfo 4
PID: 4
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 5
PID: 5
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ _


Sep 2 22:47
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/xv6-public
Error: Process with PID 3 not found.
12341550$ pinfo 4
PID: 4
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 5
PID: 5
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 6
PID: 6
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 1
PID: 1
Name: init
State: SLEEPING
Memory Size: 12288
12341550$ pinfo 2
PID: 2
Name: sh
State: SLEEPING
Memory Size: 16384
12341550$ pinfo 10
Error: Process with PID 10 not found.
12341550$ pinfo 9
Error: Process with PID 9 not found.
12341550$ pinfo 8
PID: 8
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$
QEMU
Machine View
PID: 6
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ pinfo 1
PID: 1
Name: init
State: SLEEPING
Memory Size: 12288
12341550$ pinfo 2
PID: 2
Name: sh
State: SLEEPING
Memory Size: 16384
12341550$ pinfo 10
Error: Process with PID 10 not found.
12341550$ pinfo 9
Error: Process with PID 9 not found.
12341550$ pinfo 8
PID: 8
Name: testproc
State: RUNNABLE
Memory Size: 12288
12341550$ _
"1.png" selected (333.8 kB)