Code Modification Report

Date System Call

Make File

```
CS333_PROJECT ?= 0
PRINT_SYSCALLS ?= 1
CS333_CFLAGS ?= -DPDX_XV6
ifeq ($(CS333_CFLAGS), -DPDX_XV6)
CS333_UPROGS += _halt _uptime
endif

ifeq ($(PRINT_SYSCALLS), 1)
CS333_CFLAGS += -DPRINT_SYSCALLS
endif

ifeq ($(CS333_PROJECT), 1)
CS333_CFLAGS += -DCS333_P1
CS333_UPROGS += _date
endif
```

user.h

```
struct stat;
struct rtcdate;
struct uproc;

// system calls
#ifdef CS333_P1
int date(struct rtcdate*);
#endif // CS333_P1
```

usys.S

```
SYSCALL(getpid)
SYSCALL(sbrk)
SYSCALL(sleep)
SYSCALL(uptime)
SYSCALL(halt)
SYSCALL(date)
```

syscall.h

```
#define SYS_close   SYS_mkdir+1
#define SYS_halt    SYS_close+1
#define SYS_date    SYS_halt+1
```

syscall.c

```
[SYS_mkdir]   sys_mkdir,
[SYS_close]   sys_close,
#ifdef PDX_XV6
[SYS_halt]    sys_halt,
#endif // PDX_XV6
#ifdef CS333_P1
[SYS_date]    sys_date,
#endif // PDX_XV6
};
```

```
 num = curproc->tf->eax;
  if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
    curproc->tf->eax = syscalls[num]();
    #ifdef PRINT_SYSCALLS
      cprintf("%s -> %d \n",syscallnames[num], num);
    #endif
  } else {
```

sysproc.c

```
int
sys_date ( void )
{
  struct rtcdate *d ;
  if (argptr ( 0 ,( void*)&d , sizeof ( struct rtcdate)) < 0)
    return -1;
  cmostime(d);
  return 0;
}
```

Process Information

proc.c

```
p->start_ticks = ticks;
```

proc.h

```
// Per-process state
struct proc {
  uint start_ticks;
  uint sz;                     // Size of process memory (bytes)
  pde_t* pgdir;
```

Syscall Tracing

syscall.c

```
syscall(void)
{
  int num;
  struct proc *curproc = myproc();

  num = curproc->tf->eax;
  if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
    curproc->tf->eax = syscalls[num]();
    #ifdef PRINT_SYSCALLS
      cprintf("%s -> %d \n",syscallnames[num], num);
    #endif
  } else {
    cprintf("%d %s: unknown sys call %d\n",
            curproc->pid, curproc->name, num);
    curproc->tf->eax = -1;
  }
}
```