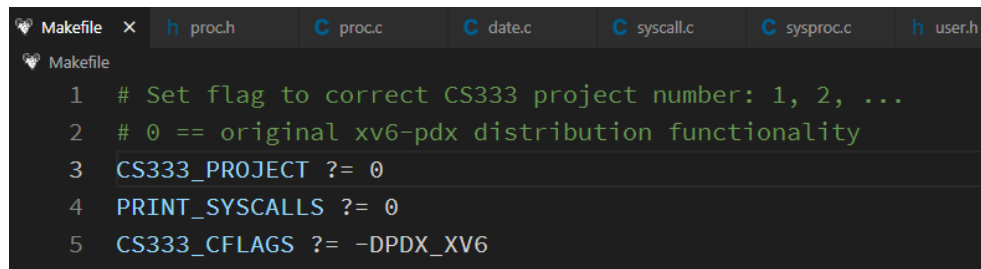


1. Compilation Test

Makefile (line 3)

a. Sebelum Perubahan

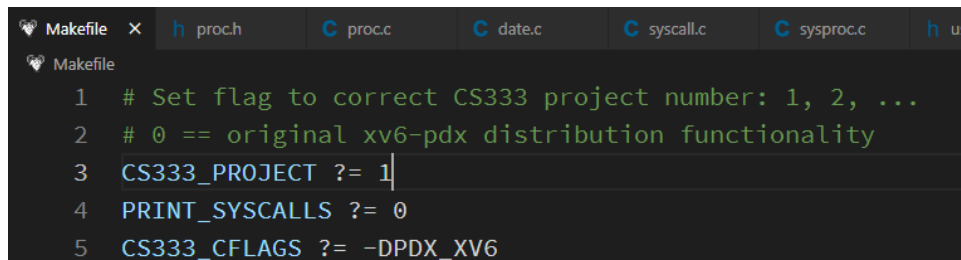


The screenshot shows a code editor with a Makefile open. The Makefile content is as follows:

```
1 # Set flag to correct CS333 project number: 1, 2, ...
2 # 0 == original xv6-pdx distribution functionality
3 CS333_PROJECT ?= 0
4 PRINT_SYSCALLS ?= 0
5 CS333_CFLAGS ?= -DPDX_XV6
```

Line 3 is highlighted, showing `CS333_PROJECT ?= 0`.

b. Setelah Perubahan



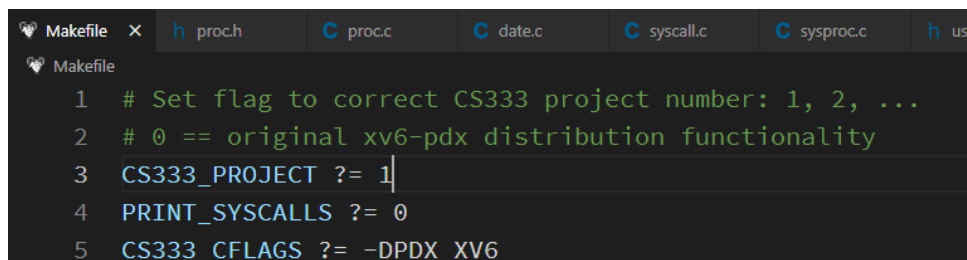
The screenshot shows the same code editor with the Makefile. Line 3 has been changed to:

```
3 CS333_PROJECT ?= 1
```

2. System Call Tracing

- Makefile (line 4)

a) Sebelum Perubahan

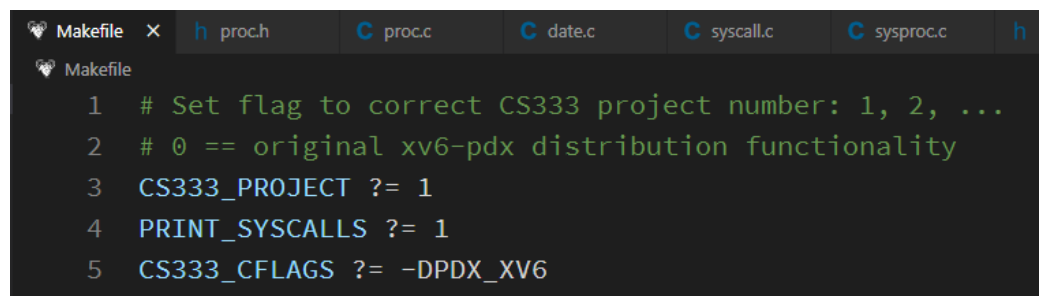


The screenshot shows a code editor with a Makefile open. The Makefile content is as follows:

```
1 # Set flag to correct CS333 project number: 1, 2, ...
2 # 0 == original xv6-pdx distribution functionality
3 CS333_PROJECT ?= 1
4 PRINT_SYSCALLS ?= 0
5 CS333_CFLAGS ?= -DPDX_XV6
```

Line 4 is highlighted, showing `PRINT_SYSCALLS ?= 0`.

b) Setelah Perubahan



The screenshot shows the same code editor with the Makefile. Line 4 has been changed to:

```
4 PRINT_SYSCALLS ?= 1
```

- Syscall.c (line 190 – 192)
 - a) Sebelum Perubahan

```

184     int num;
185     struct proc *curproc = myproc();
186
187     num = curproc->tf->eax;
188     if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
189         curproc->tf->eax = syscalls[num]();
190     }
191     else {
192         cprintf("%d %s: unknown sys call %d\n",
193             curproc->pid, curproc->name, num);
194         curproc->tf->eax = -1;
195     }
196 }
  
```

- b) Setelah Perubahan

```

187     num = curproc->tf->eax;
188     if(num > 0 && num < NELEM(syscalls) && syscalls[num]) {
189         curproc->tf->eax = syscalls[num]();
190         #ifdef PRINT_SYSCALLS
191             cprintf("%s -> %d \n", syscallnames[num], curproc->tf->eax);
192         #endif
193     } else {
194         cprintf("%d %s: unknown sys call %d\n",
195             curproc->pid, curproc->name, num);
196         curproc->tf->eax = -1;
197     }
198 }
  
```

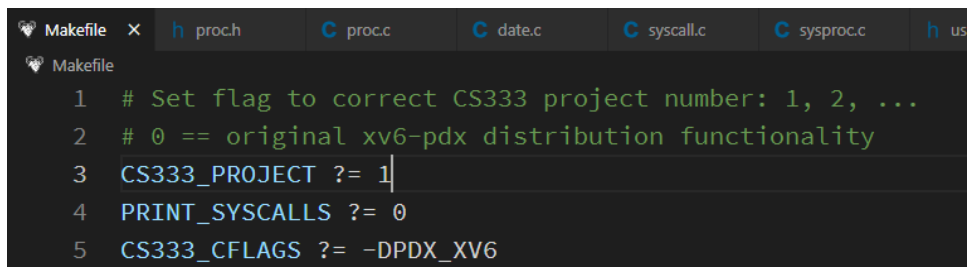
3. Conditional Compilation

- Makefile
 - a) Pada saat CS333_PROJECT != 0

```

1 # Set flag to correct CS333 project number: 1, 2, ...
2 # 0 == original xv6-pdx distribution functionality
3 CS333_PROJECT != 0
4 PRINT_SYSCALLS != 0
5 CS333_CFLAGS += -DPDX_XV6
  
```

b) Pada saat CS333_PROJECT != 1

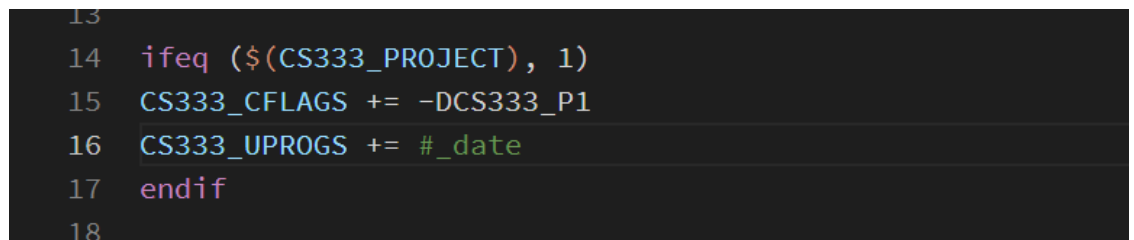


```
Makefile x h proc.h C proc.c C date.c C syscall.c C sysproc.c h us
Makefile
1 # Set flag to correct CS333 project number: 1, 2, ...
2 # 0 == original xv6-pdx distribution functionality
3 CS333_PROJECT != 1
4 PRINT_SYSCALLS != 0
5 CS333_CFLAGS += -DPDX_XV6
```

4. Date System Call

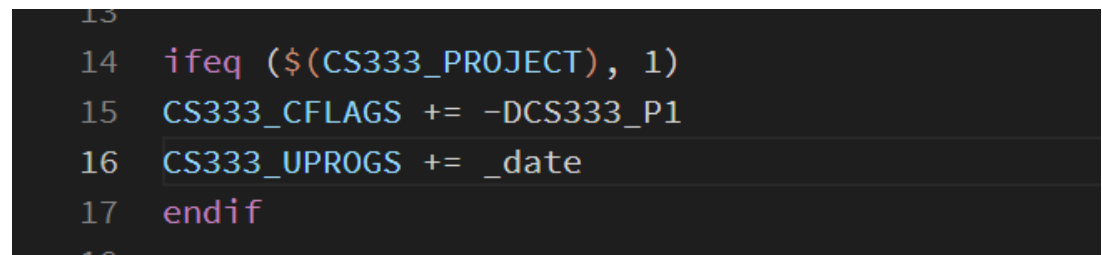
- Makefile (line 16)

a) Sebelum Perubahan



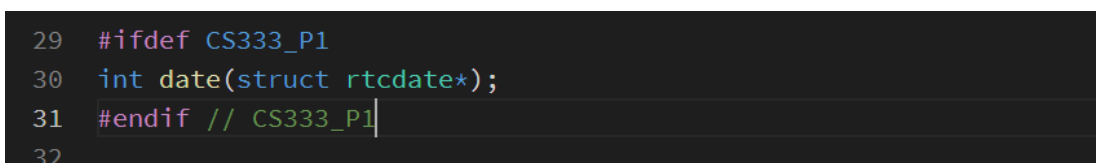
```
13
14 ifeq ($(CS333_PROJECT), 1)
15 CS333_CFLAGS += -DCS333_P1
16 CS333_UPROGS += #_date
17 endif
18
```

b) Setelah perubahan



```
13
14 ifeq ($(CS333_PROJECT), 1)
15 CS333_CFLAGS += -DCS333_P1
16 CS333_UPROGS += _date
17 endif
18
```

- user.h (penambahan pada line 29-31)



```
29 #ifdef CS333_P1
30 int date(struct rtcdate*);
31 #endif // CS333_P1
32
```

- usys.S (Penambahan SYSCALL(date) pada line 33)

```
23 SYSCALL(fstat)
24 SYSCALL(link)
25 SYSCALL(mkdir)
26 SYSCALL(chdir)
27 SYSCALL(dup)
28 SYSCALL(getpid)
29 SYSCALL(sbrk)
30 SYSCALL(sleep)
31 SYSCALL(uptime)
32 SYSCALL(halt)
33 SYSCALL(date)
34
```

- syscall.h (Penambahan (#define SYS_date SYS_halt+1) pada line 24)

```
15 #define SYS_uptime SYS_sleep+1
16 #define SYS_open SYS_uptime+1
17 #define SYS_write SYS_open+1
18 #define SYS_mknod SYS_write+1
19 #define SYS_unlink SYS_mknod+1
20 #define SYS_link SYS_unlink+1
21 #define SYS_mkdir SYS_link+1
22 #define SYS_close SYS_mkdir+1
23 #define SYS_halt SYS_close+1
24 #define SYS_date SYS_halt+1
25 // student system calls begin here. Follow the existing pattern.
26
```

- syscall.c

a) menambahkan SYS_date pada line 141 - 143

```
135 [SYS_close] sys_close,
136 #ifdef PDX_XV6
137 [SYS_halt] sys_halt,
138 #endif // PDX_XV6
139
140 //Adding new SYS_date
141 #ifdef CS333_P1
142 [SYS_date] sys_date,
143 #endif // CS333_P1
144
145 };
146
```

b) Menambahkan SYS_date pada line 175 - 177

```
168 [SYS_mkdir] "mkdir",
169 [SYS_close] "close",
170 #ifdef PDX_XV6
171 [SYS_halt] "halt",
172 #endif // PDX_XV6
173
174 //Adding new SYS_date
175 #ifdef CS333_P1
176 [SYS_date] "date",
177 #endif // CS333_P1
178 };
179 #endif // PRINT_SYSCALLS
180
```

- sysproc.c (menambahkan fungsi sys_date(void) pada line 101 - 112)

```
101 int
102 sys_date(void)
103 {
104     struct rtcdate *d;
105
106     if(argptr(0, (void*)&d, sizeof(struct rtcdate)) < 0)
107         return -1;
108     else{
109         cmostime(d);
110         return 0;
111     }
112 }
113
```

5. Process Information

- proc.h (menambahkan uint start_ticks; pada line 52)

```
43  uint pid;                // Process ID
44  struct proc *parent;     // Parent process. NULL indicates no parent
45  struct trapframe *tf;    // Trap frame for current syscall
46  struct context *context; // swtch() here to run process
47  void *chan;              // If non-zero, sleeping on chan
48  int killed;              // If non-zero, have been killed
49  struct file *ofile[NOFILE]; // Open files
50  struct inode *cwd;        // Current directory
51  char name[16];           // Process name (debugging)
52  uint start_ticks;
53 };
54
```

- proc.c

a) line 152

```
144  sp -= 4;
145  *(uint*)sp = (uint)trapret;
146
147  sp -= sizeof *p->context;
148  p->context = (struct context*)sp;
149  memset(p->context, 0, sizeof *p->context);
150  p->context->eip = (uint)forkret;
151
152  p->start_ticks = ticks;
153
154  return p;
155 }
156
```

b) Line 568 - 584

```
564 #elif defined(CS333_P1)
565 void
566 procdumpP1(struct proc *p, char *state_string)
567 {
568     int digunakan_s;
569     int digunakan_ms = ticks - (p -> start_ticks);
570     digunakan_s = digunakan_ms/1000;
571     digunakan_ms = digunakan_ms % 1000;
572
573     char* mulai= "";
574     if(digunakan_ms < 100 && digunakan_ms > 10){
575         mulai = "0";
576     }
577     if(digunakan_ms < 10){
578         mulai = "00";
579     }
580
581     cprintf("%d\t%s\t%s%d.%s%d\t%s\t%d\t",
582         p->pid, p->name, " ",
583         digunakan_s, mulai, digunakan_ms,
584         states[p->state], p->sz);
585     return;
586 }
587 #endif
588
```