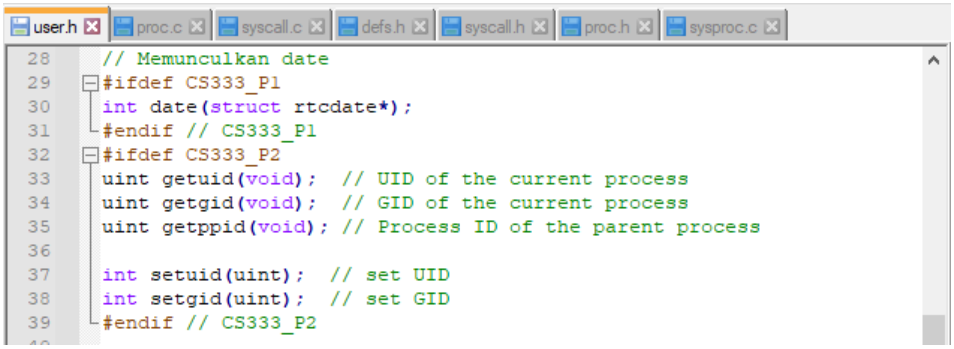
ASSIGNMENT 2

Nama : Atikah Khonsa Salsabila

NIM : 1313619002

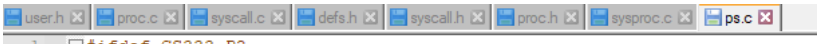
Prodi : Ilmu Komputer 2019

1. File user.h line 28-39




```
28  // Memunculkan date
29  #ifdef CS333_P1
30  int date(struct rtcdate*);
31  #endif // CS333_P1
32  #ifdef CS333_P2
33  uint getuid(void);  // UID of the current process
34  uint getgid(void);  // GID of the current process
35  uint getppid(void); // Process ID of the parent process
36
37  int setuid(uint);  // set UID
38  int setgid(uint);  // set GID
39  #endif // CS333_P2
```

2. File ps.c line 1-56




```
1  #ifdef CS333_P2
2  #include "types.h"
3  #include "user.h"
4  #include "uproc.h"
5
6  #define MAX 16
7
8  int
9  main(void)
10 {
11   struct uproc *proc = malloc(sizeof(struct uproc)*MAX);
12   int proc_num = getprocs(MAX, proc);
13   printf(1,"PID\tName\t\tUID\tGID\tPPID\tElapsed\tCPU\tState\tSize\n
14
15   int i;
16   for(i = 0; i<proc_num; i++){
17     struct uproc current_proc = proc[i];
18     uint elapsed_ticks = current_proc.elapsed_ticks;
19     uint elapsed_s = elapsed_ticks/1000;
20     uint elapsed_ms = elapsed_ticks%1000;
21
22     uint elapsed_cpu_ticks = current_proc.CPU_total_ticks;
23     uint elapsed_cpu_s = elapsed_cpu_ticks/1000;
24     uint elapsed_cpu_ms = elapsed_cpu_ticks % 1000;
25
26     char* zero = "";
27     if(elapsed_ms < 100 && elapsed_ms >= 10)
28       zero = "0";
29     if(elapsed_ms < 10)
30       zero = "00";
31
32     char* cpu_zero = "";
33     if(elapsed_cpu_ms < 100 && elapsed_cpu_ms >= 10)
34       cpu_zero = "0";
35     if(elapsed_cpu_ms < 10)
36       cpu_zero = "00";
```

```
32      char* cpu_zero = "";
33      if(elapsed_cpu_ms < 100 && elapsed_cpu_ms >= 10)
34        cpu_zero = "0";
35      if(elapsed_cpu_ms < 10)
36        cpu_zero = "00";
37
38      printf(
39        1,
40        "%d\t%s\t\t%d\t%d\t%d\t%d.%s%d\t%d.%s%d\t%s\t%d\n",
41        current_proc.pid,
42        current_proc.name,
43        current_proc.uid,
44        current_proc.gid,
45        current_proc.ppid,
46        elapsed_s, zero, elapsed_ms,
47        elapsed_cpu_s, cpu_zero, elapsed_cpu_ms,
48        current_proc.state,
49        current_proc.size
50      );
51    }
52
53    free(proc);
54    exit();
55  }
56  #endif
```

3. File proc.c line 9-11

user.h | proc.c | syscall.c | defs.h | syscall.h | proc.h

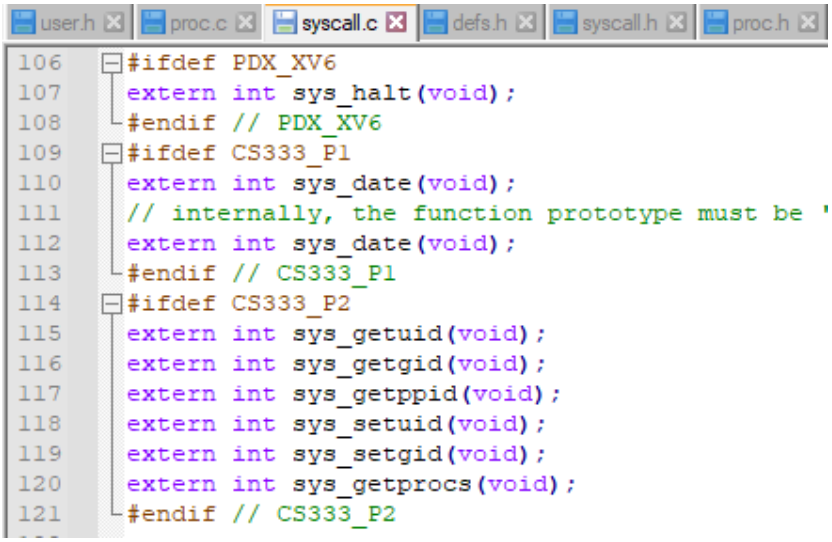
```
1   #include "types.h"
2   #include "defs.h"
3   #include "param.h"
4   #include "memlayout.h"
5   #include "mmu.h"
6   #include "x86.h"
7   #include "proc.h"
8   #include "spinlock.h"
9   #ifdef CS333_P2
10  #include "uproc.h"
11  #endif
```

4. File syscall.c line 107

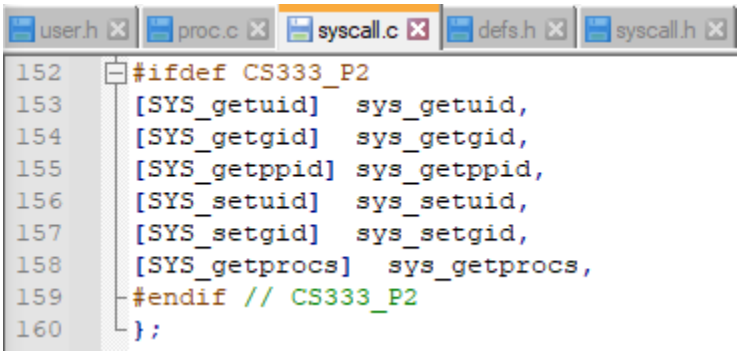
user.h | proc.c | syscall.c | defs.h

```
106 #ifdef PDX_XV6
107 extern int sys_halt(void);
```

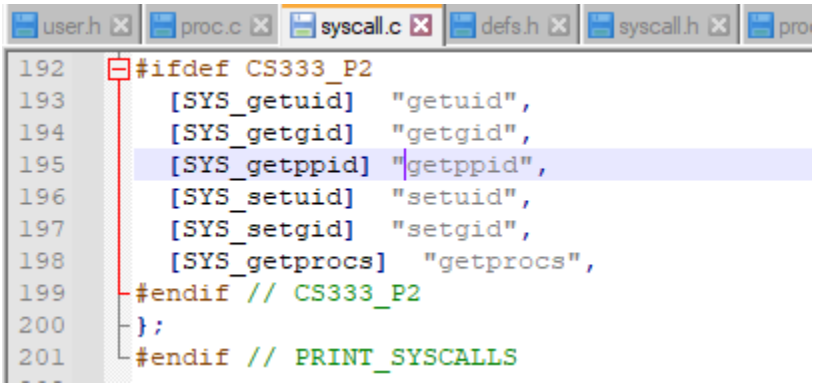
5. File syscall.c line 110 dan 112-119



6. File syscall.c line 152-159



7. File syscall.c line 192-199



8. File defs.h line 1-3

user.h | proc.c | syscall.c | defs.h

```
1  #ifdef CS333_P2
2  #include "uproc.h"
3  #endif
```

9. File defs.h line 130-132

user.h | proc.c | syscall.c | defs.h | syscall.h | proc.h | sysproc.c | ps.

```
130 #ifdef CS333_P2
131 int             getprocs(uint max, struct uproc* upTable);
132 #endif // CS333_P2
```

10. File syscall.h line 25-30

```
25  #define SYS_date    SYS_halt+1  // project 1
26  #define SYS_getuid  SYS_date+1 //project2
27  #define SYS_getgid  SYS_getuid+1
28  #define SYS_getppid SYS_getgid+1
29  #define SYS_setuid  SYS_getppid+1
30  #define SYS_setgid  SYS_setuid+1
31  #define SYS_getprocs    SYS_setgid+1
```

11. File proc.h line 54-59

```
54  #ifdef CS333_P2 //project2
55    uint uid;
56    uint gid;
57    uint cpu_ticks_total;
58    uint cpu_ticks_in;
59  #endif // CS333_P2
```

12. File sysproc.c line 115-170

```
115 #ifdef CS333_P2
116 int
117 sys_getuid(void)
118 {
119   return myproc()->uid;
120 }
121
122 int
123 sys_getgid(void)
124 {
125   return myproc()->gid;
126 }
127
128 int
129 sys_getppid(void)
130 {
131   if(myproc()->parent == NULL)
132     return myproc()->pid;
133   else
134     return myproc()->parent->pid;
135 }
136
137 int
138 sys_setuid(void)
139 {
140   int test;
141   if(argint(0, &test)<0)
142     return -1;
143   if(test < 0 || test >32767)
144     return -1;
145   else{
146     myproc()->uid = test;
147     return 0;
148   }
149 }
150
151 int
```

```
151 int
152 sys_setgid(void)
153 {
154   int test;
155   if(argint(0, &test)<0)
156     return -1;
157   if(test < 0 || test >32767)
158     return -1;
159   else{
160     myproc()->gid = test;
161     return 0;
162   }
163 }
164
165 int
166 sys_getprocs(void)
167 {
168   struct uproc *p;
169   int max;
170
171   if(argint(0,&max)<0){
172     return -1;
173   }
174   if(argptr(1, (void*)&p, sizeof(struct uproc) * max) < 0)
175     return -1;
176   return getprocs(max, p);
177 }
178 #endif // CS333_P2
```

13. File time.c line 1-46

```c
#ifdef CS333_P2
#include "types.h"
#include "user.h"

int main(int argc, char *argv[]){
    if(argc == 1) {
      printf(1, "(null) ran in 0.00\n");
    } else {
      int start = uptime();
      int pid = fork();

      if (pid > 0) {
        pid = wait();
      } else if (pid == 0) {
        exec(argv[1], argv+1);
        printf(1, "ERROR: Unknown Command\n");
        kill(getppid());
        exit();
      } else {
        printf(1, "ERROR: Fork error return -1\n");
      }

      int end = uptime();
      int timelapse = end - start;
      int seconds = timelapse/1000;
      int ms = timelapse%1000;
      char *msZeros = "";

      if (ms < 10) {
        msZeros = "00";
      } else if (ms < 100) {
        msZeros = "0";
      }

      printf(
        1,
        "%s ran in %d.%s%d\n",
        argv[1],
        seconds,
        msZeros,
        ms
      );
    }
    exit();
}
#endif // CS333_P2
```

14. File usys.S line 34-39

```
SYSCALL(getuid) #project2
SYSCALL(getgid)
SYSCALL(getppid)
SYSCALL(setuid)
SYSCALL(setgid)
SYSCALL(getprocs)
```

15. File testsetuid.c line 1-11

```c
#ifdef CS333_P2
#include "types.h"
#include "user.h"

int
main(int argc, char *argv[])
{
  printf(1, "***** In %s: my uid is %d\n\n", argv[0], getuid());
  exit();
}
#endif
```