” به نام خداوند جان و خرد”

فاز اول پروژه سیستم عامل
گزارشکار

نام دانشجو: کیانا آقاکثیری 9831006

نام استاد : دکتر جوادی

تاریخ تحویل پروژه : 1400/07/30

ابتدا یک تابع getProcCount در proc.c می سازیم. در این تابع ، روی کل processها حرکت میکند و به ازای آنهایی که UNUESD نباشند شمارنده را یکی زیاد میکند و در نهایت آن را چاپ کرده و return میکند.

```
int getProcCount(void){
    struct proc *p;
    int counter=0;
    for(p = ptable.proc; p < &ptable.proc[NPROC]; p++){
      if(p->state != UNUSED) counter++;
    }
    cprintf("%d",counter);

  return counter;
}
```

سپس در فایل های دیگر آن را اضافه میکنیم تا این system call را بشناسند. فایل های دیگر :

1.MakeFile
2.defs.h
3.syscall.h
4.sysfile.c
5.sysproc.c
6.user.h
7.usys.S
8.syscall.c

حال برای اضافه کردن system call دوم تابع getReadCount را در sysproc.c تعریف میکنیم و یک متغیر extern میسازیم که این تابع آن را چاپ و return میکند.

```
extern int readCount;
int sys_getReadCount(void)
{
  cprintf("%d",readCount);
  return readCount;
}
```

حال در فایل sysfile.c تابعی به نام sys_read وجود دارد. متغیر readCount را در بیرون این تابع هم تعریف میکنیم و در این تابع میزنیم یکی اضافه شود. بدین صورت تعداد دفعاتی که فراخوانی سیستمی ***Read***توسط هر پردازه ي دیگر کاربر فراخوانی شده است (از زمانی که کرنل بوت شده)را برمیگرداند .

```c
int readCount = 0;
int
sys_read(void)
{
  readCount++;
  struct file *f;
  int n;
  char *p;

  if(argfd(0, 0, &f) < 0 || argint(2, &n) < 0 || argptr(1, &p, n) < 0)
    return -1;
  return fileread(f, p, n);
}
```

و دوباره در فایل های دیگر آن را اضافه میکنیم تا این system call را بشناسند.

سپس یک فایل تست برای آنها مینویسیم و درستی این system call را بررسی میکنیم.

getProcCountTest.c:

```c
C getProcCountTest.c
1   #include "types.h"
2   #include "stat.h"
3   #include "user.h"
4
5   int main (void) {
6       getProcCount();
7       printf(1,"proc count success");
8       exit();
9   }
```

getReadCountTest.c:

```c
C getReadCountTest.c
1   #include "types.h"
2   #include "stat.h"
3   #include "user.h"
4
5   int main (void) {
6       getReadCount();
7       printf(1,"read count success");
8       exit();
9   }
```

بدین صورت میبینیم پاسخ صحیح است: