

تنها کاری که باید در این قسمت انجام دهیم، تغییر scheduler مربوط به xv6 است، (ما صف ها را با استفاده از مقادیر مربوط به priority ها پیاده سازی کردیم، عدد 1 برای صف اول، عدد 2 برای صف دوم و اعداد بزرگتر از 2 برای صف سوم هستند. برای این که به مرور از کاهش اولویت فرآیند ها جلوگیری کنیم، باقیمانده اولویت بر 20 را گرفته ایم):

[illegible]

```

        {
            result=p1;
        }
    }
    else
    {
        result=p1;
    }
}

if(result!=0)
    result->priority=(result->priority)%20+1;
f=result;

if(f!=0)
{
    p=f;
}
if(p->state!=RUNNABLE)
    continue;
c->proc = p;
switchvm(p);
p->state = RUNNING;

swtch(&(c->scheduler), p->context);
switchkvm();

// Process is done running for now.
// It should have changed its p->state before coming back.
c->proc = 0;
}

release(&ptable.lock);
}
}

```

سپس برای امتحان کردن برنامه نوشته شده باید دستورات زیر را اجرا کنیم:

make clean

make qemu

بعد از اجرای این دستورات shell مربوط به سیستم عامل xv6 باز می‌شود و ما می‌توانیم دستور جدید و برنامه خود را تست کنیم.

وقتی که shell باز شد باید دستور زیر را در آن وارد کنیم:

prs

نمونه‌هایی از خروجی‌های برنامه در زیر آورده شده است:

```
$ prs
2
2
1
diff:1
diff2:15
2
```

```
$ prs
1
2
1
2
diff:2
diff2:16
```

با صدا کردن فراخوانی های سیستمی خود xv6 هم به مشکلی نخوردیم:

```
init: starting sh
$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
cat        2 3 16388
echo       2 4 15240
forktest   2 5 9544
grep       2 6 18608
init       2 7 15828
kill       2 8 15268
ln         2 9 15124
ls         2 10 17752
mkdir      2 11 15368
rm         2 12 15344
sh         2 13 27980
stressfs   2 14 16260
usertests  2 15 67364
wc         2 16 17120
zombie     2 17 14936
wx         2 18 16976
prs        2 19 17160
console    3 20 0
$
```