MYPRIO

2017029325 권순우
2017029716 박혜정
2017029798 성원석

구현방법 – 자료구조

```
struct sched_myprio_entity {
        struct list_head run_list;
        unsigned int aging;
        unsigned int stride;
        unsigned int  pass;
};
```

stride값이 작을수록 우선순위가 높다.

구현방법

aging = stride/100 을 반올림한 것
비율로 aging을 시켜주기 위함.
모든 task가 최대 100번 안에 다시 실행되게 하기 위함.

```
void init_task_myprio(struct task_struct *p)
{
        struct sched_myprio_entity *se = &p->myprio;

        //stride will be 10~1000(int)
        p->sched_class = &myprio_sched_class;
        if((se->stride % 100) < 50)
                se->aging = (se->stride)/100;
        else      //round aging by stride
                se->aging = (se->stride)/100 + 1 ;
        //any task must be executed before 100 upadates
        se->pass = se->stride;
}
```

구현방법

update_curr_myprio()

0. 현재 실행중인 task의 pass를 stride만큼 올려줌

1. 현재 실행되지 않는 task들의 pass값을 aging만큼 줄여줌

2. 마지막에 resched_curr를 실행

```
//not only update curr, but also aging the other tasks.
static void update_curr_myprio(struct rq *rq)
{
        struct myprio_rq *myprio_rq = &rq->myprio;
        struct list_head *myprio_queue = &myprio_rq->queue;

        struct sched_myprio_entity *se = NULL;
        struct task_struct *p = NULL;

        //int i;

        printk(KERN_INFO "      ***[MYPRIO] update_curr: update_curr starting!\n");

        ((rq->curr)->myprio).pass += ((rq->curr)->myprio).stride;

        int i=0;
        for(i;i<myprio_rq->nr_running;i++) {
                myprio_queue = myprio_queue->next;
                se = container_of(myprio_queue, struct sched_myprio_entity, run_list);
                p = container_of(se, struct task_struct, myprio);

                if(p->pid != (rq->curr)->pid) {
                        //if se->pass be less than 0, set 0.
                        if((se->pass - se->aging) < 0)
                                se->pass = 0;
                        se->pass -= se->aging;      //decrease other tasks' pass by aging to
increase priority
                }
                printk(KERN_INFO "        ***[MYPRIO] update_curr: curr->pid=%d, p->pid=%d,
pass=%u, stride=%u\n", (rq->curr)->pid ,p->pid, se->pass, se->stride);
        }
        //we don't change the order of rq because sorting every update is more difficult.
        //for every update, we must pick minimum-pass task.
        resched_curr(rq);

        printk(KERN_INFO "      ***[MYPRIO] update_curr: update_curr end!\n");
}
```

구현방법

pick_next_task_myprio()

0. queue안의 task들 중 pass값이 가장 작은 것을 next_task로 한다.

1. pass값이 같을 경우 우선순위를 따른다.

```
struct task_struct *pick_next_task_myprio(struct rq *rq, struct task_struct *prev)
{
        struct sched_myprio_entity *se = NULL;
        struct task_struct *next_p = NULL;
        struct sched_myprio_entity *next_se = NULL;

        struct myprio_rq *myprio_rq = &rq->myprio;
        struct list_head *myprio_queue = &myprio_rq->queue;

        if(myprio_rq->nr_running == 0) {
                return NULL;
        }

        if(prev->sched_class != &myprio_sched_class) {
                //printk(KERN_INFO "    ***[MYPRIO] pick_next_task: other class came in.. p
  rev->pid=%d\n", prev->pid);
                put_prev_task(rq, prev);
        }

        //select the minimum-pass task in rq
        //if the minimum-pass tasks are more than 2,
        //select the one which has the least stride.
        int i;
        unsigned int minimum_pass;
        //minimum_pass = ((rq->curr)->myprio).pass;
        unsigned int minimum_stride;
        //minimum_stride = ((rq->curr)->myprio).stride;
        //next_p = rq->curr;
        next_se = container_of(myprio_queue->next, struct sched_myprio_entity, run_list);
        next_p = container_of(next_se, struct task_struct, myprio);
        minimum_pass = next_se->pass;
        minimum_stride = next_se->stride;
        for(i=0;i<myprio_rq->nr_running;i++) {
                myprio_queue = myprio_queue->next;
                se = container_of(myprio_queue, struct sched_myprio_entity, run_list);
                if(se->pass < minimum_pass)
                {
                        minimum_pass = se->pass;
                        minimum_stride = se->stride;
                        next_se = se;
                }
                else if(se->pass == minimum_pass)
                {
                        if(se->stride < minimum_stride) {
                                minimum_stride = se->stride;
                                next_se = se;
                        }
                }
                                                                142,1-8       70%
```

aging이 되지 않는 경우

```
104003 Jun 11 22:16:13 2019os kernel: [   49.947675]     ***[MYPRIO] pick_next_task: cpu=1, prev->pid=1974, next_p->pid=1975, next_pass=360, next_stride=20, nr_running=10
104004 Jun 11 22:16:13 2019os kernel: [   49.948500]           ***[MYPRIO] update_curr: update_curr starting!
104005 Jun 11 22:16:13 2019os kernel: [   49.948566]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1983  pass=512,  stride=512
104006 Jun 11 22:16:13 2019os kernel: [   49.948625]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1982  pass=999,  stride=999
104007 Jun 11 22:16:13 2019os kernel: [   49.948679]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1981  pass=500,  stride=500
104008 Jun 11 22:16:13 2019os kernel: [   49.948728]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1980  pass=377,  stride=377
104009 Jun 11 22:16:13 2019os kernel: [   49.948773]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1979  pass=400,  stride=50
104010 Jun 11 22:16:13 2019os kernel: [   49.948813]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1978  pass=450,  stride=150
104011 Jun 11 22:16:13 2019os kernel: [   49.948849]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1977  pass=400,  stride=100
104012 Jun 11 22:16:13 2019os kernel: [   49.948883]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1976  pass=360,  stride=40
104013 Jun 11 22:16:13 2019os kernel: [   49.948913]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1975  pass=380,  stride=20
104014 Jun 11 22:16:13 2019os kernel: [   49.948941]                   ***[MYPRIO] update_curr: curr->pid=1975, p->pid=1974  pass=370,  stride=10
104015 Jun 11 22:16:13 2019os kernel: [   49.948983]           ***[MYPRIO] update_curr: update_curr end!
104016 Jun 11 22:16:13 2019os kernel: [   49.949056]     ***[MYPRIO] pick_next_task: cpu=1, prev->pid=1975, next_p->pid=1976, next_pass=360, next_stride=40, nr_running=10
104017 Jun 11 22:16:13 2019os kernel: [   49.949505]           ***[MYPRIO] update_curr: update_curr starting!
104018 Jun 11 22:16:13 2019os kernel: [   49.949570]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1983  pass=512,  stride=512
104019 Jun 11 22:16:13 2019os kernel: [   49.949628]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1982  pass=999,  stride=999
104020 Jun 11 22:16:13 2019os kernel: [   49.949682]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1981  pass=500,  stride=500
104021 Jun 11 22:16:13 2019os kernel: [   49.949747]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1980  pass=377,  stride=377
104022 Jun 11 22:16:13 2019os kernel: [   49.949791]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1979  pass=400,  stride=50
104023 Jun 11 22:16:13 2019os kernel: [   49.949830]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1978  pass=450,  stride=150
104024 Jun 11 22:16:13 2019os kernel: [   49.949867]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1977  pass=400,  stride=100
104025 Jun 11 22:16:13 2019os kernel: [   49.949900]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1976  pass=400,  stride=40
104026 Jun 11 22:16:13 2019os kernel: [   49.949930]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1975  pass=380,  stride=20
104027 Jun 11 22:16:13 2019os kernel: [   49.949957]                   ***[MYPRIO] update_curr: curr->pid=1976, p->pid=1974  pass=370,  stride=10
104028 Jun 11 22:16:13 2019os kernel: [   49.949982]           ***[MYPRIO] update_curr: update_curr end!
104029 Jun 11 22:16:13 2019os kernel: [   49.950057]     ***[MYPRIO] pick_next_task: cpu=1, prev->pid=1976, next_p->pid=1974, next_pass=370, next_stride=10, nr_running=10
104030 Jun 11 22:16:13 2019os kernel: [   49.950471]           ***[MYPRIO] update_curr: update_curr starting!
104031 Jun 11 22:16:13 2019os kernel: [   49.950534]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1983  pass=512,  stride=512
104032 Jun 11 22:16:13 2019os kernel: [   49.950590]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1982  pass=999,  stride=999
104033 Jun 11 22:16:13 2019os kernel: [   49.950641]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1981  pass=500,  stride=500
104034 Jun 11 22:16:13 2019os kernel: [   49.950687]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1980  pass=377,  stride=377
104035 Jun 11 22:16:13 2019os kernel: [   49.950729]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1979  pass=400,  stride=50
104036 Jun 11 22:16:13 2019os kernel: [   49.950767]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1978  pass=450,  stride=150
104037 Jun 11 22:16:13 2019os kernel: [   49.950802]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1977  pass=400,  stride=100
104038 Jun 11 22:16:13 2019os kernel: [   49.950833]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1976  pass=400,  stride=40
104039 Jun 11 22:16:13 2019os kernel: [   49.950862]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1975  pass=380,  stride=20
104040 Jun 11 22:16:13 2019os kernel: [   49.950888]                   ***[MYPRIO] update_curr: curr->pid=1974, p->pid=1974  pass=380,  stride=10
104041 Jun 11 22:16:13 2019os kernel: [   49.950912]           ***[MYPRIO] update_curr: update_curr end!
104042 Jun 11 22:16:13 2019os kernel: [   49.950995]     ***[MYPRIO] pick_next_task: cpu=1, prev->pid=1974, next_p->pid=1980, next_pass=377, next_stride=377, nr_running=10
104043 Jun 11 22:16:13 2019os kernel: [   49.951119]           ***[MYPRIO] update_curr: update_curr starting!
104044 Jun 11 22:16:13 2019os kernel: [   49.951122]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1983  pass=512,  stride=512
104045 Jun 11 22:16:13 2019os kernel: [   49.951125]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1982  pass=999,  stride=999
104046 Jun 11 22:16:13 2019os kernel: [   49.951128]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1981  pass=500,  stride=500
104047 Jun 11 22:16:13 2019os kernel: [   49.951130]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1980  pass=754,  stride=377
104048 Jun 11 22:16:13 2019os kernel: [   49.951133]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1979  pass=400,  stride=50
104049 Jun 11 22:16:13 2019os kernel: [   49.951136]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1978  pass=450,  stride=150
104050 Jun 11 22:16:13 2019os kernel: [   49.951139]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1977  pass=400,  stride=100
104051 Jun 11 22:16:13 2019os kernel: [   49.951141]                   ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1976  pass=400,  stride=40
                                                                                                                                                                104003,1      98%
```

aging이 되는 경우(1) – update 초반

```
***[MYPRIO] update_curr: update_curr starting!
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1980  pass=492, stride=512
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1979  pass=959, stride=999
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1978  pass=480, stride=500
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1977  pass=361, stride=377
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1976  pass=46, stride=50
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1975  pass=142, stride=150
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1974  pass=96, stride=100
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1973  pass=40, stride=40
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1972  pass=40, stride=20
        ***[MYPRIO] update_curr: curr->pid=1971, p->pid=1971  pass=50, stride=10
***[MYPRIO] update_curr: update_curr end!
***[MYPRIO] pick_next_task: cpu=1, prev->pid=1971, next_p->pid=1972, next_pass=40, next_stride=20, nr_running=10
***[MYPRIO] update_curr: update_curr starting!
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1980  pass=487, stride=512
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1979  pass=949, stride=999
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1978  pass=475, stride=500
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1977  pass=357, stride=377
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1976  pass=45, stride=50
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1975  pass=140, stride=150
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1974  pass=95, stride=100
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1973  pass=40, stride=40
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1972  pass=60, stride=20
        ***[MYPRIO] update_curr: curr->pid=1972, p->pid=1971  pass=50, stride=10
***[MYPRIO] update_curr: update_curr end!
***[MYPRIO] pick_next_task: cpu=1, prev->pid=1972, next_p->pid=1973, next_pass=40, next_stride=40, nr_running=10
***[MYPRIO] update_curr: update_curr starting!
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1980  pass=482, stride=512
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1979  pass=939, stride=999
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1978  pass=470, stride=500
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1977  pass=353, stride=377
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1976  pass=44, stride=50
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1975  pass=138, stride=150
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1974  pass=94, stride=100
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1973  pass=80, stride=40
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1972  pass=60, stride=20
        ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1971  pass=50, stride=10
***[MYPRIO] update_curr: update_curr end!
```

aging이 되는 경우(2) – 우선순위가 낮은 것이 실행된 순간

```
        ***[MYPRIO] update_curr: update_curr starting!
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1980  pass=252, stride=512
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1979  pass=479, stride=999
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1978  pass=240, stride=500
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1977  pass=550, stride=377
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1976  pass=304, stride=50
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1975  pass=350, stride=150
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1974  pass=250, stride=100
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1973  pass=280, stride=40
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1972  pass=260, stride=20
                ***[MYPRIO] update_curr: curr->pid=1973, p->pid=1971  pass=250, stride=10
        ***[MYPRIO] update_curr: update_curr end!
***[MYPRIO] pick_next_task: cpu=1, prev->pid=1973, next_p->pid=1978, next_pass=240, next_stride=500, nr_running=10
        ***[MYPRIO] update_curr: update_curr starting!
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1980  pass=247, stride=512
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1979  pass=469, stride=999
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1978  pass=740, stride=500
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1977  pass=546, stride=377
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1976  pass=303, stride=50
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1975  pass=348, stride=150
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1974  pass=249, stride=100
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1973  pass=280, stride=40
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1972  pass=260, stride=20
                ***[MYPRIO] update_curr: curr->pid=1978, p->pid=1971  pass=250, stride=10
        ***[MYPRIO] update_curr: update_curr end!
***[MYPRIO] pick_next_task: cpu=1, prev->pid=1978, next_p->pid=1980, next_pass=247, next_stride=512, nr_running=10
        ***[MYPRIO] update_curr: update_curr starting!
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1980  pass=759, stride=512
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1979  pass=459, stride=999
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1978  pass=735, stride=500
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1977  pass=542, stride=377
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1976  pass=302, stride=50
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1975  pass=346, stride=150
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1974  pass=248, stride=100
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1973  pass=280, stride=40
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1972  pass=260, stride=20
                ***[MYPRIO] update_curr: curr->pid=1980, p->pid=1971  pass=250, stride=10
        ***[MYPRIO] update_curr: update_curr end!
***[MYPRIO] pick_next_task: cpu=1, prev->pid=1980, next_p->pid=1974, next_pass=248, next_stride=100, nr_running=10
```

ova파일 링크

- https://drive.google.com/open?id=16HNYytJGG0dr42jFQ2w0__kQp9RJJqMg

감사합니다