171250535 蔡明卫

一、 进程饿死问题的解决

1- 读者优先的情况下

添加一个全局变量 readsum，每次读操作完成进程就会将 readsum 加 1，同时每次进程开始读之前，都会检查 readsum 的值，如果大于设定的值（当前为 2），那么将不进入等待读的序列，这样保证读进程读一定次数后，读进程将无法继续排队，这是，写进程就可以获得执行权，与此同时，写进程将 readsum 置为 0，那么写进程执行后读进程又可以进入等待序列，这样保证读者优先的情况下，写进程也不至于永远饿死。

```
if(readsum>=2){
continue;
}
```

进入读者排队序列前检查 readsum 的值

```
system_disp_str(name);
system_disp_str(" stop reading\n");

readsum+=1;
```

每次读结束后将 readsum 加 1

```
PUBLIC void read(char* name,int time){
int temptime=time;
while(1){
temptime=time;
if(readsum>=2){
continue;
}
system_P(w);
system_P(rmutex);
if(readcount==0)system_P(rw);
readcount++;
system_disp_str(name);
system_disp_str(" begin reading\n");
state=0;
system_V(rmutex);

while(temptime!=0){
system_disp_str(name);
system_disp_str(" is reading\n");
system_process_sleep(2000);

temptime--;
}
system_disp_str(name);
system_disp_str(" stop reading\n");

readsum+=1;

system_V(w);
system_process_sleep(1);
system_P(rmutex);
readcount--;
if(readcount==0){system_V(rw);}
system_V(rmutex);
system_process_sleep(10);
}
}
```

```
PUBLIC void write(char* name,int time){
int temptime=time;

while(1){
temptime=time;
system_P(rw);
system_disp_str(name);
system_disp_str(" begin writing\n");
state=1;
readsum=0;
//system_process_sleep(2000*time);
while(temptime!=0){
system_disp_str(name);
system_disp_str(" is writing\n");
system_process_sleep(2000);

temptime--;
}


system_disp_str(name);
system_disp_str(" stop writing\n");
system_V(rw);


}
}
```

写进程执行后就将 readsum 置为 0，保证后续读进程能继续排队。

2-写者优先实现缓解饿死问题的思路同上。