

Quarta esercitazione di SOPLR

Riprendendo il programma randwalk.cc, realizzare e documentare il seguente programma: riscrivere randwalk in forma di multithreading. Cioe' realizzare due thread con il codice (il codice è solo abbozzato...)

=====primo thread

```
for (;;)
{
    down(s)
    down(s1);
    newspeed=avoidspeed;
    if(sp->GetScan(1)+sp->GetScan(15)<sp->GetScan(7)+sp->GetScan(8))
    newturnrate = -turnrate; else newturnrate = turnrate;
    pp.SetSpeed(newspeed, newturnrate);
    up(s1);
}
```

=====secondo thread

```
for (;;)
{
    down(s1);
    avoidcount = 0;
    newspeed = speed;
    if(!randcount) //rotazione random ogni 2 secondi
    {
        randint = rand() % 41 - 20; //numero random tra -20 e 20
        newturnrate = dtor(randint);
        randcount = 20;
    }
    randcount--;
    pp.SetSpeed(newspeed, newturnrate);
    up(s1);
}
```

Si ricorda che i sensori [2][3][4][5] sono frontali, [10][11][12][13] sono posteriori, [1][15] sono a sinistra e [7][8] a destra del robot, e un angolo di jog negativo sterza a destra e positivo sterza a sinistra.

Il main utilizza il codice

```
for (;;)
{
    robot.Read();
    if( ((sp->GetScan (2) < minfrontdistance) ||
        (sp->GetScan (3) < minfrontdistance) ||
        (sp->GetScan (4) < minfrontdistance) ||
        (sp->GetScan (5) < minfrontdistance) ))
    up(s);
}
```

I thread ed i semafori sono di Posix1 e devono essere definiti. I semafori s, s1 sono inizializzati rispettivamente a 0 e 1.