void kernel(vector<task>&S, vector<task>&SN, int ktar, task vtar, Matrix<int, 2>&G, Matrix<int, 2>&ta,int ts, int tc, int tr)

{

unsigned int i;

int m;

int t;

t = ts + tc + tr;

vector<task>re;

re = S;

for (i = 0; i < re.size(); i++)

if (vtar.num == re[i].num)

{

re[i].chan = ktar;

if (ktar == 3)

{

re[i].FTl = 0;

re[i].RTl = 0;

}

}

while (re.size()!=0)

{

get\_ready1(re, G);

get\_ready2(re);

m = 0;

while ((re[m].ready1!=0)&&(re[m].ready2 != 0))

m = m + 1;

if (re[m].chan == 3)

{

re[m].FTWR = clouds(re[m], SN, G, ts, tc, tr);

re[m].ST = re[m].FTWR - t;

}

else

{

re[m].FTl = localse(re[m], SN, G, ta);

re[m].ST = re[m].FTl - ta(re[m].num - 1, re[m].chan);

}

SN.push\_back(re[m]);

re.erase(re.begin() + m);

}

}