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;

vector<int>m;

int t;

unsigned int a,b;

t = ts + tc + tr;

vector<task>re;

vector<task>stack;

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);

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

if ((re[i].ready1 == 0) && (re[i].ready2 == 0))

{

stack.push\_back(re[i]);

m.push\_back(i);

}

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

{

if (stack[i].chan == 3)

{

stack[i].FTWR = clouds(stack[i], SN, G, ts, tc, tr);

stack[i].ST = stack[i].FTWR - t;

}

else

{

stack[i].FTl = localse(stack[i], SN, G, ta);

stack[i].ST = stack[i].FTl - ta(stack[i].num - 1, stack[i].chan);

}

SN.push\_back(stack[i]);

}

stack.erase(stack.begin(), stack.end());

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

{

m[i] = m[i] - i;

re.erase(re.begin() + m[i]);

}

m.erase(m.begin(), m.end());

}

}