

交通量シュミレーションのプログラム

```
0001: /*
0002: traffic.c
0003: yukio sugawa
0004: 2012/03/07
0005: セルオートマトンに基づく交通量シュミレーション
0006: ルールと初期状態から、時間発展を計算します。
0007: 使い方 ./traffic (流入量) < (初期ファイル名)
0008: */
0009:
0010: #include <stdio.h>
0011: #include <stdlib.h>
0012: #include <string.h>
0013:
0014: #define N 50
0015: #define R 8
0016: #define MAXT 50
0017: #define BUFSIZE 1024
0018: #define RULE "184"
0019: #define VEHICLE "-"
0020:
0021: void setrule(char *arg, int rule[]);
0022: void initca(int ca[]);
0023: int cvalue(char ch);
0024: void putca(int ca[]);
0025: void nextt(int ca[], int rule[]);
0026:
0027: int main(int argc, char *argv[])
0028: {
0029:     int t;
0030:     int ca[N] = {0};
0031:     int rule[R];
0032:     int flowrate = MAXT;
0033:
0034:     if(argc < 2){
0035:         fprintf(stderr, "使い方 traffic (流入量) < (初期フ
0036:         iles名)\n");
0037:         exit(1);
0038:     }
0039:     flowrate = atoi(argv[1]);
0040:     if(flowrate <= 0){
0041:         fprintf(stderr, "流入率が正しくありません (%d) \n", flowrate);
0042:         exit(1);
0043:     }
0044:
0045:     setrule(RULE, rule);
0046:
0047:     initca(ca);
```

```

0048: printf("t= 0 ");
0049: putca(ca);
0050:
0051: for(t = 1; t < MAXT; ++t){
0052:     nextt(ca, rule);
0053:     if((t % flowrate) == 0){
0054:         ca[N - 2] = 1;
0055:     }
0056:     printf("t=%3d", t);
0057:     putca(ca);
0058: }
0059: return 0;
0060: }
0061:
0062: void nextt(int ca[], int rule[])
0063: {
0064:     int nextca[N] = {0};
0065:     int i;
0066:
0067:     for(i = 1; i < N - 1; i++){
0068:         nextca[i] = rule[ca[i + 1] * 4 + ca[i] * 2 + ca[i - 1]];
0069:     }
0070:
0071:     for(i = 0; i < N; ++i){
0072:         ca[i] = nextca[i];
0073:     }
0074: }
0075:
0076: void putca(int ca[])
0077: {
0078:     int i;
0079:
0080:     for(i = N - 1; i >= 0; --i){
0081:         if(ca[i] == 1){
0082:             printf(VEHICLE);
0083:         }else{
0084:             printf(" ");
0085:         }
0086:     }
0087:     printf("\n");
0088: }
0089:
0090: void initca(int ca[])
0091: {
0092:     char linebuf[BUFSIZE];
0093:     int i = 0;
0094:
0095:     if(fgets(linebuf, BUFSIZE, stdin) == NULL){
0096:         fprintf(stderr, "初期値の読み取りに失敗しました\n");

```

```

0097:     exit(1);
0098: }
0099:
0100: for(i = 0; linebuf[i] != '\0'; ++i){
0101:     ca[N - 1 - i] = cvalue(linebuf[i]);
0102: }
0103: }
0104:
0105: int cvalue(char ch)
0106: {
0107:     if(ch == '1'){
0108:         return 1;
0109:     }else{
0110:         return 0;
0111:     }
0112: }
0113:
0114: void setrule(char *arg, int rule[])
0115: {
0116:     int ruleno;
0117:     int i;
0118:
0119:     ruleno = atoi(arg);
0120:     if((ruleno < 0) || (255 < ruleno)){
0121:         fprintf(stderr, "ルール番号が正しくありません (%d) \n", ruleno);
0122:         exit(1);
0123:     }
0124:     printf("ルール番号 %d :", ruleno);
0125:
0126:     for(i = 0; i < R; ++i){
0127:         rule[i] = ruleno % 2;
0128:         ruleno /= 2;
0129:     }
0130:
0131:     for(i = R - 1; i >= 0; --i){
0132:         printf("%1d", rule[i]);
0133:     }
0134:     printf("\n");
0135: }

```

初期状態ファイル 「trafficinit.txt」の内容

```
111111111111111111
```

渋滞している所に、流入量を 2 と指定することで、2 時刻ごとに 1 台の自動車を流入させる例です。

「traffic」の操作方法

```
./traffic 2 < trafficinit.txt 「Enter」
```

上記の「./」は、Linux でのプログラムを実行する時の例です。
ウィンドウズでは、traffic 2 < trafficinit.txt と入力してください。

実行結果

```
ルール番号 184 :10111000  
t= 0 -----  
t= 1 ----- -  
t= 2 ----- - -  
t= 3 ----- - - -  
t= 4 ----- - - - -  
t= 5 ----- - - - - -  
t= 6 ----- - - - - - -  
t= 7 ----- - - - - - - -  
t= 8 ----- - - - - - - - -  
t= 9 ----- - - - - - - - - -  
t= 10 ----- - - - - - - - - - -  
t= 11 ----- - - - - - - - - - - -  
t= 12 ----- - - - - - - - - - - - -  
t= 13 ----- - - - - - - - - - - - - -  
t= 14 ----- - - - - - - - - - - - - - -  
t= 15 ----- - - - - - - - - - - - - - - -  
t= 16 ----- - - - - - - - - - - - - - - - -
```