

## OpenMP で並列処理

```
001: /*
002:  * Array のインクリメント (OpenMP 版)
003:  */
004:
005: #include<stdio.h>
006: #include<stdlib.h>
007: #include<omp.h>
008: #define N 16
009:
010: int main (int argc, char *argv[])
011: {
012:     int i;
013:     int *rootBuf;
014:     int num_of_threads;
015:
016:     if(argc!=2){
017:         printf("usage: a.out <number of threads>\n");
018:         return(1);
019:     }
020:
021:     num_of_threads = atoi(argv[1]);
022:     omp_set_num_threads(num_of_threads);
023:
024:     rootBuf = (int *)malloc(N * sizeof(int));
025:
026:     /* 配列 Initialize */
027:     for(i=0;i<N;i++){
028:         rootBuf[i] = i;
029:     }
030:
031:     /* 並列処理の開始 */
032: #pragma omp parallel
033:     printf("Exec by thread 0 (total 0 threads)\n",omp_get_thread_num(), omp_get_num_threads());
034: #pragma omp for
035:     /* Increment */
036:     for (i = 0; i < N; i++) {
037:         rootBuf[i] = rootBuf[i] + 1;
038:     }
039:
040:     /* 演算結果の出力 */
041:     printf("\n");
042:     for (i = 0; i < N; i++) printf("rootbuf[%d] = %d\n", i, rootBuf[i]);
043:
044:     /* 終了処理 */
045:     free(rootBuf);
046:
047:     return(0);
048: }
```

## コンパイル

```
gcc -fopenmp list3.c
```