

# HW0 Part II: Exchange Problem

**Course:** ENEE159V/H  
**Title:** Exchange Problem  
**Date Assigned:** January 30, 2009  
**Date Due:** February 12, 2009

## 1 Problem statement

**Input:** An array  $A = A(0), \dots, A(n-1)$  and another array  $B = B(0), \dots, B(n-1)$  of elements (any kind).

**Task:** The exchange problem is to swap all elements of A and B. For example if  $A = (1, 5, 7, 3)$  and  $B = (2, 6, 4, 8)$ , after exchange  $A = (2, 6, 4, 8)$  and  $B = (1, 5, 7, 3)$ .

Serial algorithm with  $3n$  operations,  $3n$  steps, 1 space ( $X$ ):

```
for i=1 to n do
  X = A[i];
  A[i] = B[i];
  B[i] = X;
```

Parallel algorithm (1) with  $3n$  operations, 3 steps,  $n$  spaces ( $X[1..n]$ ):

```
for i=1 to n pardo
  X[i] = A[i];
  A[i] = B[i];
  B[i] = X[i];
```

Parallel algorithm (2) with  $4n$  operations, 2 steps,  $2n$  spaces ( $X[1..n]$ ,  $Y[1..n]$ ):

```
for i=1 to n pardo
  X[i] = A[i]; | B[i] = X[i];
  Y[i] = B[i]; | A[i] = Y[i];
```

## 2 Assignment

- Write XMTc programs realizing serial and both parallel algorithms.
- Discuss: Is the second parallel algorithm preferred to the first one?

## 3 The program

You should name your XMTc files as follows: exchange.s.c (serial), exchange1.p.c (parallel 1), exchange2.p.c (parallel 2).

### 3.1 Setting up the environment

The header files and the binary files can be downloaded from the course website. To get the data files, enter the following commands from your Linux environment:

```
$ wget http://terpconnect.umd.edu/~jspeiser/homework0ii.tgz
$ tar xzvf homework0ii.tgz
```

This will create the directory *exchange* with following folders: *data*, *src*, and *doc*. Data files are available in data directory.

### 3.2 Input format

You are given two arrays *A* and *B* that contains *n* integers.

int n	The size of the arrays
int A[n]	The array A
int B[n]	The array B

### 3.3 Data sets

Run all your programs (serial and parallel) using the following data files. You can directly include the header file into your XMTC code with *#include* or you can include the header file with the compile option *-include*. To run the compiled program you will need to specify the binary data with *-binload* option. When you are ready to collect clock cycle information, be sure to add the *-cycle* option to the *xmtsim* command.

Data Set	Header File	Binary file
n = 16	data/16/exchange.h	data/16/exchange.xbo
n = 3200	data/3200/exchange.h	data/3200/exchange.xbo
n = 16000	data/16000/exchange.h	data/16000/exchange.xbo
n = 64000	data/64000/exchange.h	data/64000/exchange.xbo

### 3.4 Output

**Prepare and fill the following table:** Create a text file named table.txt in *doc*. Remove any printf statements from your code while taking these measurements. Printf statements increase the clock and instruction count. Therefore the measurements with printf statements and may not reflect the actual time and work done. Is the second parallel algorithm preferred to the first one? After having filled the table, write your arguments.

Input size	n = 16	n = 3200	n = 16000	n = 64000
Serial Clock Cycles				
Parallel1 Clock Cycles				
Parallel2 Clock Cycles				

### 3.5 Command Examples

When developing, generally use:

```
$ xmtcc exchange.s.c -include ../data/64000/exchange.h ../data/exchange.xbo -o myprogram  
$ xmtsim myprogram.sim -binload myprogram.b -loadbase 0
```

When ready to collect cycle counts use simply add the `-cycle` option as mentioned above.

### 3.6 Submission

For this project, all work should be contained within the exchange directory. Upon completing the project use the following command: `tar czvf homework0ii.tgz exchange`. Otherwise, follow all the instructions for project submission listed the course website: <http://terpconnect.umd.edu/~jspeiser/enee159vh/#submit>