

BE OpenMP

L. Giraud* F. Couderc[†]

2018-2019

1 `exo1.c`

Compiler le code source `exo1.c` via la commande `make exo1`, exécuter le code avec 1 thread (`export OMP_NUM_THREADS=1`) puis avec 4 threads (`export OMP_NUM_THREADS=4`). Modifier le source de sorte que

1. la valeur en sortie de la boucle de la variable `count` soit égale à la valeur de la variable `n`.
2. instrumenter le code de sorte que la variable `mycount` permette de comptabiliser le nombre de boucles exécutées par chacun des threads.

2 `exo2.c`

Compiler le code source `exo2.c` via la commande `make exo2`, exécuter le code en faisant varier le nombre threads pour observer un dysfonctionnement possible.

1. Expliquer le dysfonctionnement observé,
2. Proposer une solution pour assurer que l'exécution parallèle soit conforme à la séquentielle.

3 `exo3.c`

Compiler le code source `exo3.c` via la commande `make exo3`, exécuter le code en faisant varier le nombre threads pour observer un dysfonctionnement possible.

1. Expliquer le dysfonctionnement observé,
2. Proposer une solution pour assurer que l'exécution parallèle soit conforme à la séquentielle.

*INRIA Bordeaux - Sud Ouest

[†]CNRS - IMT