# COMP 7500/7506 Lecture 18 Project 3-6 AUbatch – PThreads

**Review Exercise 1:** What is the difference between the following two code segments?

for (row = 0; row < n; row++)

for (column = 0; column < n; column++)

c[row][column] =

dot\_product( get\_row(a, row),

get\_col(b, col));

for (row = 0; row < n; row++)

for (column = 0; column < n; column++)

c[row][column] =

create\_thread( dot\_product(get\_row(a, row), get\_col(b, col)));

**Exercise 2:** A thread can terminate in three ways. Can you list two ways of terminating threads? (Hint: We just learned these two)

void \*thread\_function(void \*arg);

char message[] = "Hello World";

int main() {

int res;

pthread\_t a\_thread;

void \*thread\_result;

res = pthread\_create(&a\_thread, NULL, thread\_function,

(void \*)message);

if (res != 0) {

perror("Thread creation failed");

exit(EXIT\_FAILURE);

}

printf("Waiting for thread to finish...\n");

res = pthread\_join(a\_thread, &thread\_result);

if (res != 0) {

perror("Thread join failed");

exit(EXIT\_FAILURE);

}

printf("Thread joined, it returned %s\n", (char \*)thread\_result);

printf("Message is now %s\n", message);

exit(EXIT\_SUCCESS);

}

void \*thread\_function(void \*arg) {

printf("thread\_function is running. Argument was %s\n",

(char \*)arg);

sleep(3);

strcpy(message, "see you. Bye!");

pthread\_exit("No, thank you for the CPU time");

}

**Exercise 3:** Read the above source code and answer the following two questions.

1. How does the parent process pass an argument to the child thread?
2. How does the child thread return a result back to the parent process?

**Exercise 4.** (1) Which variable is a shared one? (2)Insert pthread\_mutex\_lock() and pthread\_mutex\_unlock() into the following code.

pthread\_mutex\_t task\_queue\_lock; int task\_available;

main() {

task\_available = 0;

pthread\_mutex\_init(&task\_queue\_lock, NULL);

....

}

void \*producer(void \*producer\_thread\_data) {

....

while (!done()) {

inserted = 0;

create\_task(&my\_task);

while (inserted == 0) {

if (task\_available == 0) {

insert\_into\_queue(my\_task);

task\_available = 1;

inserted = 1;

}

}

}

}

**Exercise 5.** Please complete the following source code using PThread conditiona variables.

void \*producer(void \*producer\_thread\_data) {

int inserted;

while (!done()) {

create\_task();

pthread\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

while (task\_available == 1)

pthread\_cond\_wait\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

insert\_into\_queue();

task\_available = 1;

pthread\_cond\_signal(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

pthread\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}