

Objectives - Tue 4/9/2019

Section 2 (12:00)

□ Unit testing

□ Parallel programming (intro)

3 A's

□ Arrange

□ Act

□ Assert

```
int test_ main () {  
    mu_start();  
    // -----  
    ARRANGE  
    eg. create_image  
    ACT  
    eg. write, crop, etc.  
    ASSERT  
    eg. check bound, content, etc.  
    // -----  
    mu_end();  
}
```

Sometimes

arrange or assert
may be minimal or
nothing.

No manual inspection required

- The tests should be able to run on their own
- This is the foundation of regression testing

Run all tests
whenever something
changes and/or
periodically (eg. nightly)

Order

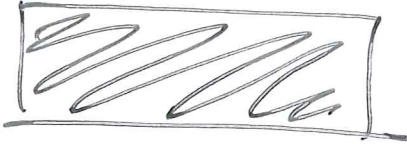
- Tests should be able to run in any order
- Every test should start with a clean slate

test_read() should not depend on test_write()

Need enough support code so each test can be independent.

create_bmp(...)
set_pixel(...)

To view image on Windows
via PuTTY and Xming

- Install + Start Xming
- display .bmp

For our purposes on HW12....

"helper function"

- ~~_____~~ ()
- Not called from outside this context (ie., file)

"support function"

- Mainly to aid your tests.
- May be checked for the HW.

path coverage

```
void f(int a) {  
    ~~~~~  
    ~~~~~  
    if (~~~~) { //A  
        ~~~~~  
    }  
    ~~~~~  
    if (~~~~) { //B  
        ~~~~~  
    }  
}
```

Note: More detail and an example was given on Thu 4/11 for the 12:30 section.

Line coverage
- Executed every line ≥ 1 x through all of your tests.

path coverage

	(A)	(B)
Enter both	Y	Y
Enter (A) only	Y	N
Enter (B) only	N	Y
Enter neither	N	N

Can't always test every path.

Bugs \neq



Your code is
flawed

run-time
error
handling



System and/or
input data is
atypical / malformed

What to test

- Easy cases
- “Edge cases” (boundary values)
- “Corner cases” (turning points)
 - Go into an IF/WHILE/FOR or not
 - Test values near the turning point.
- Special cases (look for "except" in spec)

Process

often an application that you started

Ex • /test_bmp
vim
bash

Separate memory space

Threads

Run within a process to allow parallel computation.

Shared memory space

split an image to save time

