### create 2 projects under 1 solutions

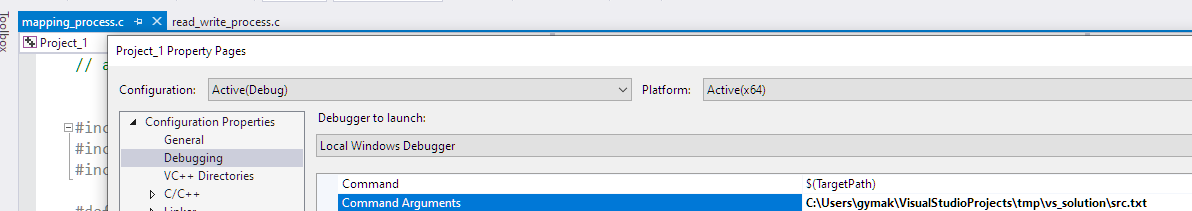
#### project 1

##### name of the source file “mapping process.c”

##### arguments: the path to the large sized file (I called it src.txt)

##### purose: creating “mapped object” on the RAM , using handle to that file (without loading this huge file to the physical memore, because it is too large)

##### this object will represent the file itself, that is located on the secondary memory (HHD)



#### project 2

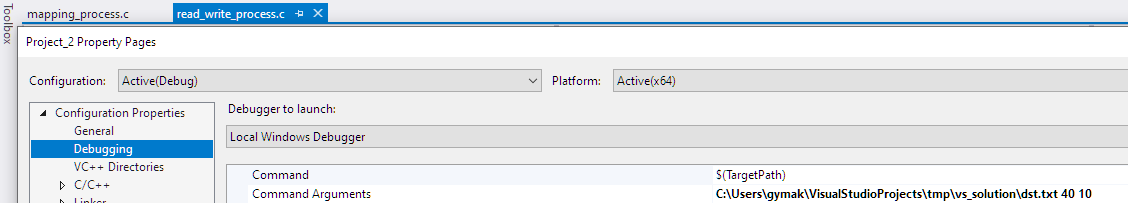
##### name of the source file “read\_write\_process.c”

##### arguments:

###### the path to the destination file which will be created using that process (I called it dst.txt)

###### amount of bytes to read or write

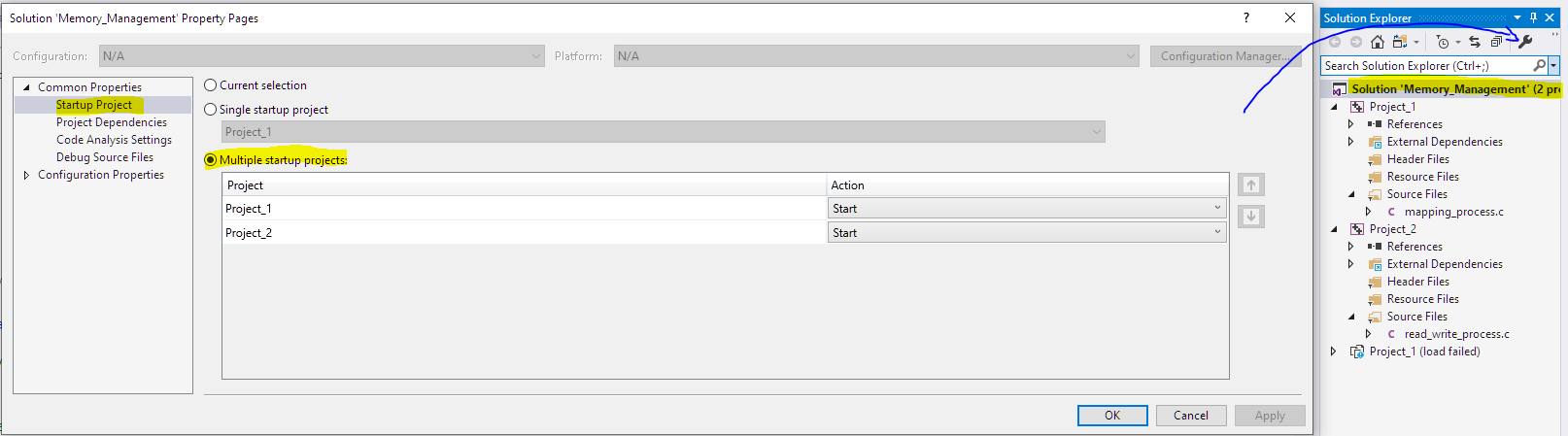
###### offset bytes



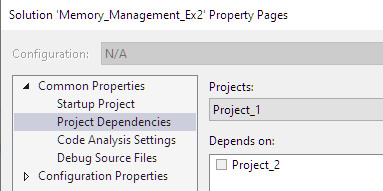
### configurations in Visual Studio

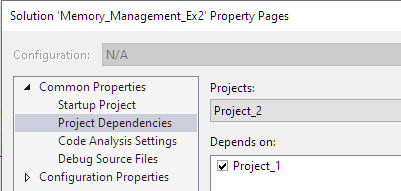
#### because of the dependencies, we must comfigure project 1 to run before project 2

#### define start up for the solution:



#### define dependecise:





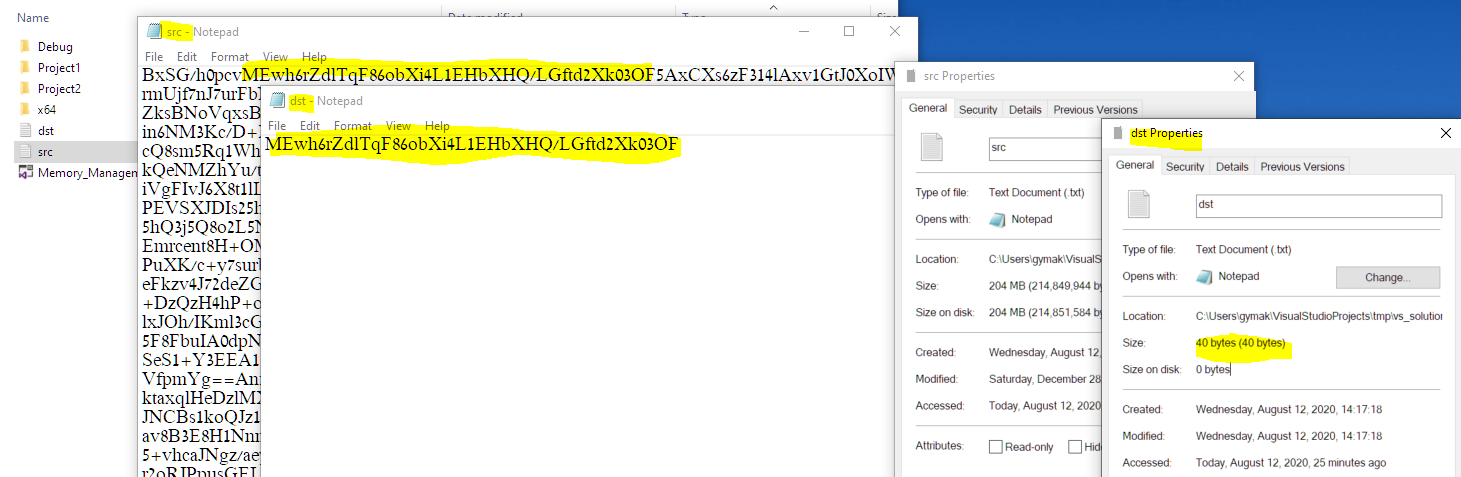
### build and run

#### choose the process 1 project (meaning choose tab of process 1)

#### click run (green arrow)

### check the created file

#### note – for demonstration and the ability to work with the large sized file, I created it with only 200 Mbyte, but the code will be the same for dealing with large files



### notes for process 1 (described at the code as well):

#### the responsibily to close the mapped object is for the process that uses it

#### MAPPED\_OBJ\_NAME cannot be changed without changing this value in other processes

### notes for process 2 (described at the code as well):

#### the responsibily to close the mapped object is for the process that uses it, in other words: this process

#### MAPPED\_OBJ\_NAME cannot be changed without changing this value in other processes and the process which created the mapped object

### For more details please refer to MSDN

#### Functions

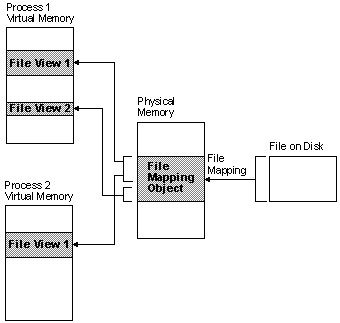
##### [CreateFileW](https://docs.microsoft.com/he-il/windows/win32/api/fileapi/nf-fileapi-createfilew)

##### [CreateFileMappingA](https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-createfilemappinga)

###### use to create “File mapping object” that associated with the large file on the HDD

###### this object named “section” sometimes

###### ilustration:



##### [MapViewOfFile](https://docs.microsoft.com/he-il/windows/win32/api/memoryapi/nf-memoryapi-mapviewoffile?redirectedfrom=MSDN)

###### creating “file mapping object” that mapps part of the file on the HDD (this object is on RAM)

###### using “file mapping object” we can access to different parts of the large sized file, using virtual address.Those sections are on the RAM memory (this is why we have virtual addresses)

##### [OpenFileMapping](https://docs.microsoft.com/he-il/windows/win32/api/winbase/nf-winbase-openfilemappinga)

##### [WriteFile](https://docs.microsoft.com/he-il/windows/win32/api/fileapi/nf-fileapi-writefile)

##### [FlushViewOfFile](https://docs.microsoft.com/he-il/windows/win32/api/memoryapi/nf-memoryapi-flushviewoffile?redirectedfrom=MSDN)

##### [CloseHandle](https://docs.microsoft.com/en-us/windows/win32/memory/closing-a-file-mapping-object)

##### UnmapViewOfFile

#### Theory:

##### [Sharing Files and Memory](https://docs.microsoft.com/en-us/windows/win32/memory/sharing-files-and-memory)

##### [file mapping](https://docs.microsoft.com/en-us/windows/win32/memory/file-mapping)

##### [system error code](https://docs.microsoft.com/en-us/windows/win32/debug/system-error-codes--0-499-)