Debugging Overlays

Overview

- Makefile switches
- Overlay files
- Problem areas

Makefile Switches

- Use /v with the psylink call
  - Allows recognition of overlays by debugger
- Use -g with the ccpsx call
  - Generates debug info for the symbol file

Overlay files

What are they?

- Pure binary code
- Assembled at a fixed address

Putting them in memory

- Normally read from CD
  - Not very feasible to debug
- Two methods for debugging purposes
  - CD emulator
  - PC fileserver functions

Using the emulator

- Run executable image file and use printf's
- DBUGPSX, use emulator as source for overlay files

PC fileserver functions

- PCopen()
  - Used to get a file handle
- PClseek()
  - Used to acquire the length of overlay file
- PCread()
  - Used to read in a file from the host PC

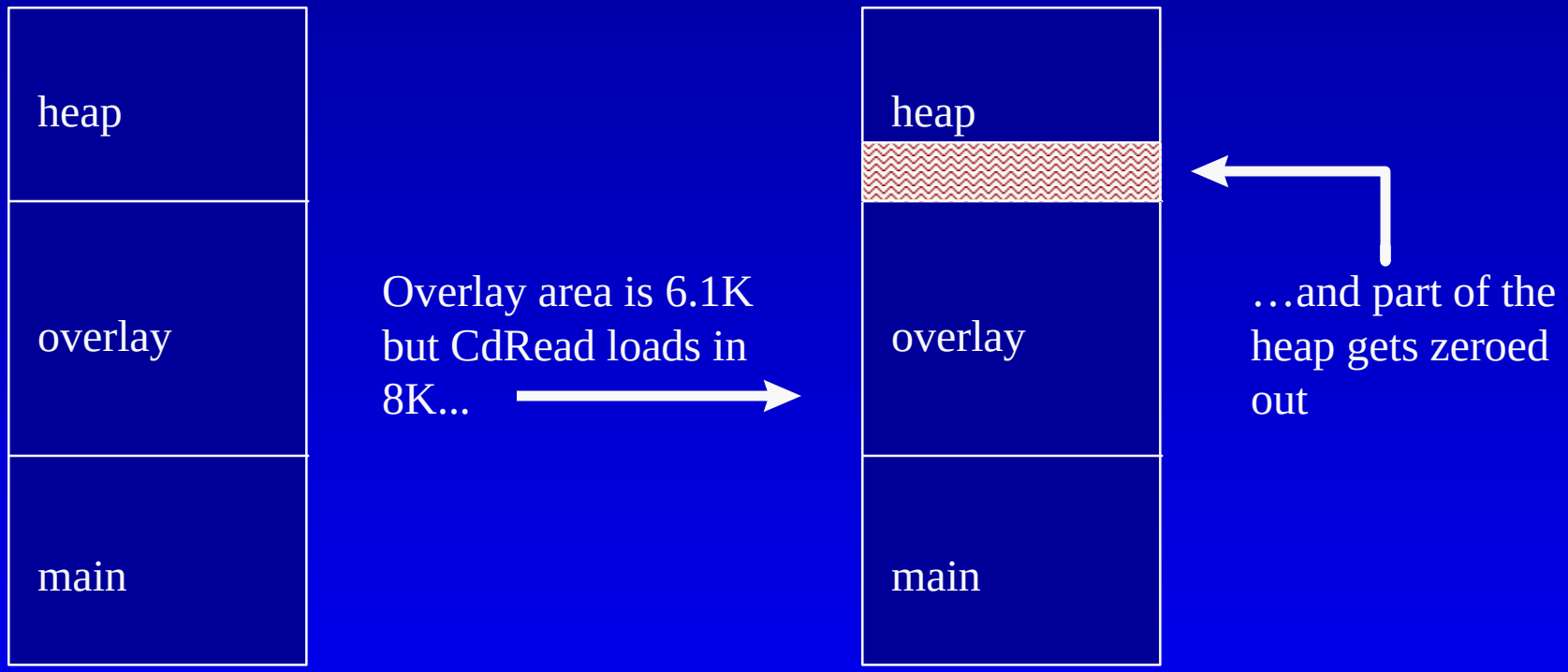
Problem Areas

- Reading the file incorrectly
- Addressing errors

Reading the file incorrectly

- Overlays need not be an integral number of sectors in length
- High level functions *CdRead()*, *CdReadFile()* read entire sectors into memory, zeroing out any bytes not present in source file

Reading the file incorrectly



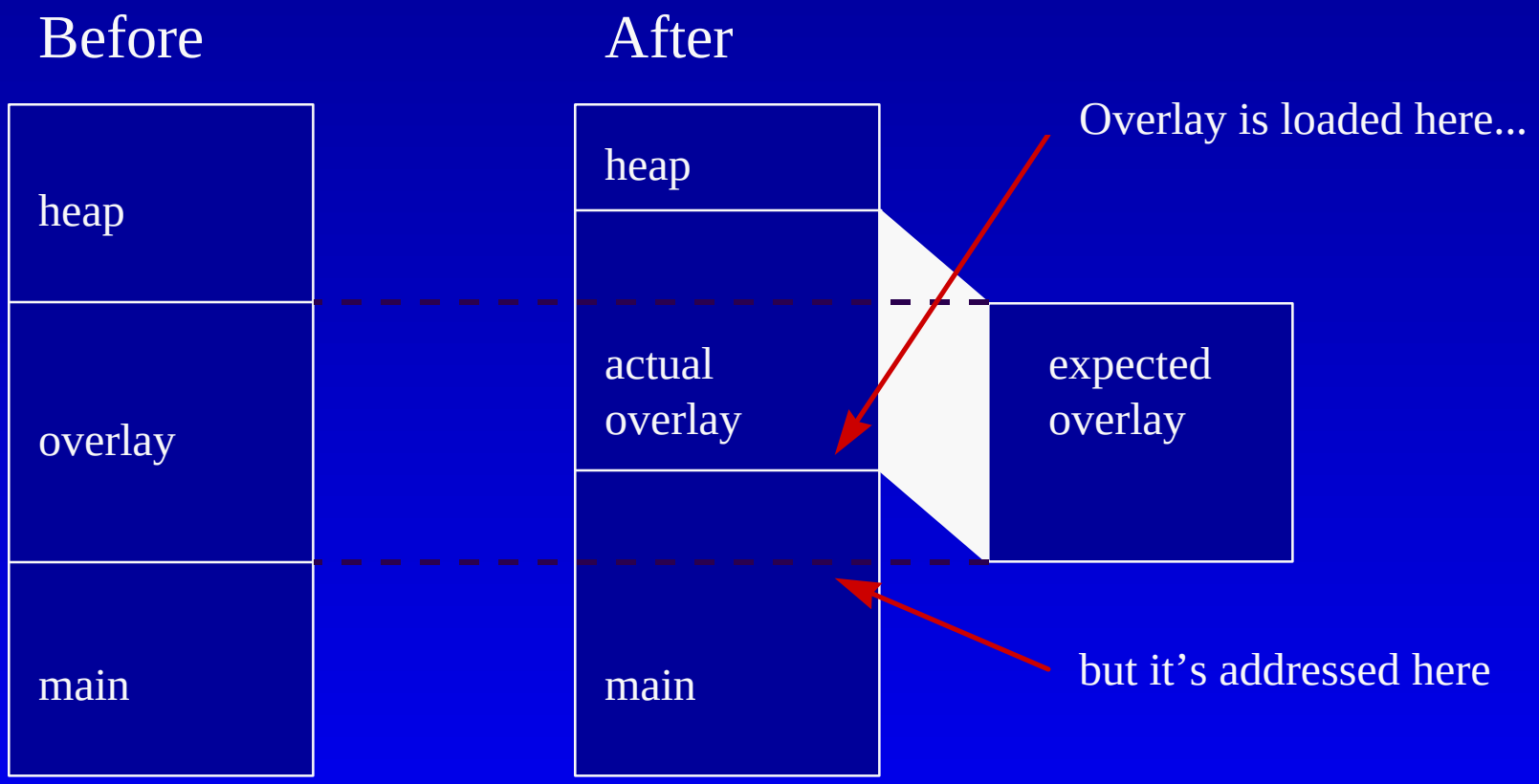
Solutions...

- Good--use low level functions, only read in what is needed
  - Best utilization of memory
- Not so good--arrange any group, heap, etc that follows the overlay so that it is out of harm's way
  - This can waste up to 2K of RAM

Addressing errors

- Code is associated with a specific address
- Even if overlay code is unchanged, changes in the parent will lead to address changes in overlays
- New main + old overlay = crash

Addressing errors



Solutions...

- Try to think of main program and overlays as one unit--where one goes, they all go
  - Especially when burning discs!
- When using *dbugpsx* with the main program and the emulator to read in overlays, remember to update your emulation image after each build