Using the Global Pointer in Overlays

Necessary Steps

‣ Evaluation

‣ Changes to the makefile

‣ Changes to the linker command file

‣ Changes to the address file

‣ Other considerations

Evaluation

‣ Speed vs. Convenience

‣ What is to be gained?

- Speed increase varies with size of overlays' sdata and sbss areas

‣ How difficult will it be?

- More overlays = More pain

Changes to the Makefile

‣ Adjust the -G setting

‣ Modify the options passed to ASPSX

Adjust the -G setting

‣ Can vary in each overlay

  • sdata and sbss should be large enough to be worthwhile

‣ Watch the 64k limit

Modify ASPSX options

‣ Normally all overlay sections end up in same group, with same prefix

- *-Wa,sprefix* generates *prefix.data, prefix.text,* etc all in a group called *prefix*

‣ However, we want the sections *unassigned to a group*

- *-Wa,s-prefix* generates the same sections, but makes no group assignments

Changes to the Linker Command File

‣ Creating additional groups

‣ Rearranging groups

‣ Placing sections in correct groups

Creating Additional Groups

‣ Each overlay will now need 3 or 4 groups

  • One for the .rdata, .data, and .text sections

  • One for the .sdata section

  • The .sbss and .bss sections may each have their own group or be combined

‣ The main .sbss section may need to be put in its own group

Creating Additional Groups

‣ Examples:

```
; for main .sbss section
msbss      group      bss

; for overlay .sdata sections
o1sdata    group      file(o1sd.bin)
o2sdata    group      over(o1sdata),file(o2sd.bin)

;for overlay .sbss and .bss sections
o1sbss     group      bss
o2sbss     group      bss,over(o1sbss)

;for the remaining overlay sections
o1text     group      file(o1t.bin)
o2text     group      over(o1text),file(o2t.bin)
```

Placing Sections in Correct Groups

‣ Assign the various overlay sections to newly created groups

- The .rdata section to the overlay text group, the .sdata section to the overlay sdata group, etc.

‣ Reassign the main .sbss section to the new main sbss group, if necessary

Placing Sections in Correct Groups

‣ Examples:
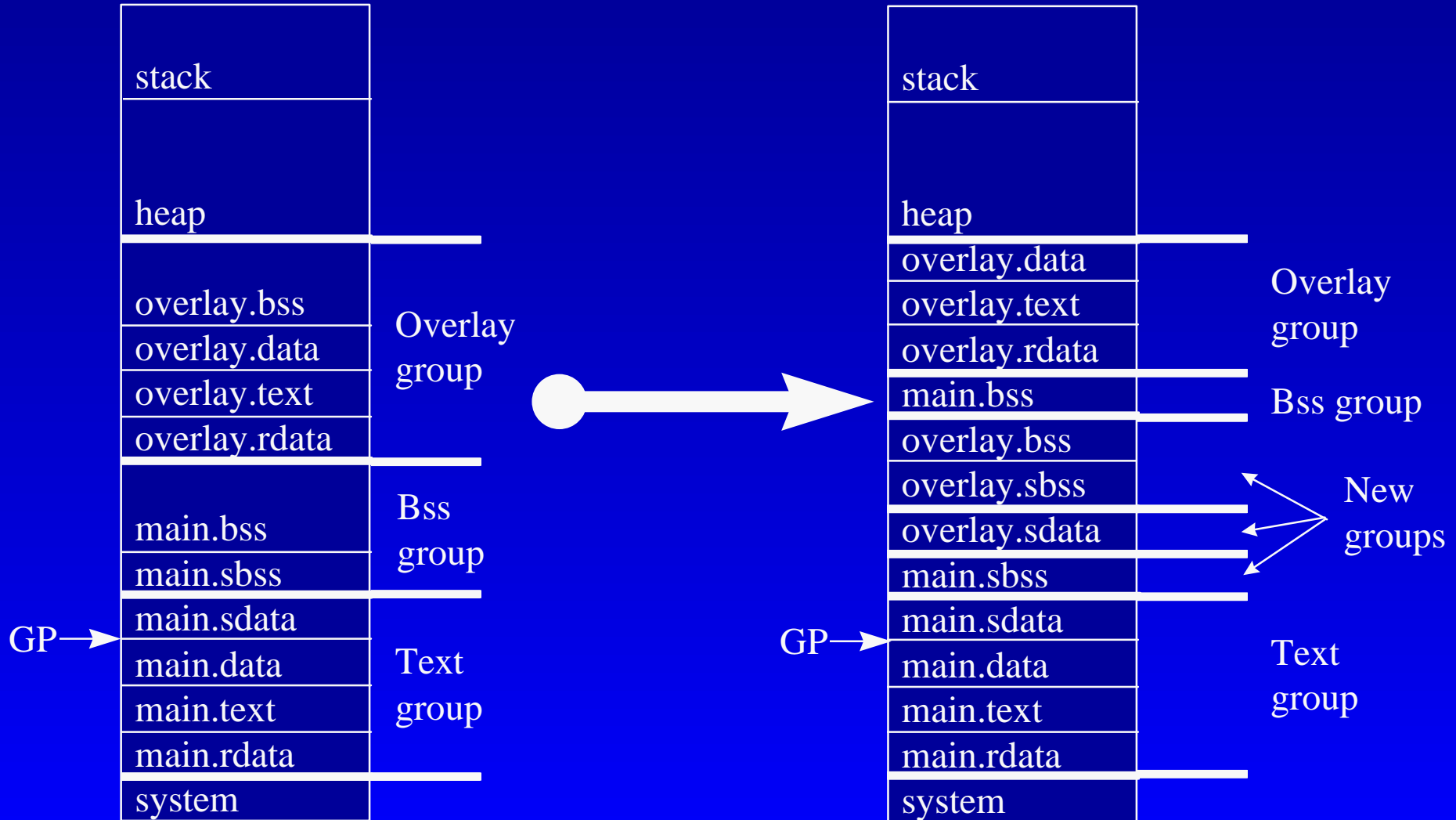
```
section    .rdata,text
section    .text,text
section    .data,text
section    .sdata,text
section    .sbss,msbss          ;put in the new  main sbss group
section    .bss,bss


section    o1.rdata,o1text
section    o1.text,o1text
section    o1.data,o1text
section    o1.sdata,o1sdata    ;put in the new overlay sdata group
section    o1.sbss,o1sbss      ;put in the new overlay sbss/bss group
section    o1.bss,o1sbss       ;put in the new overlay sbss/bss group
```

Rearranging Groups

‣ Order groups so that the various small data sections will reside in a contiguous piece of memory

- Put the overlay sdata group after the main text or sbss group

- Put the overlay sbss group after the overlay sdata group

- Put the main sbss group after the overlay sbss group, if necessary

Rearranging Groups

stack	
heap	
overlay.bss	Overlay group
overlay.data	
overlay.text	
overlay.rdata	
main.bss	Bss group
main.sbss	
main.sdata	GP →
main.data	Text group
main.text	
main.rdata	
system	

stack	
heap	
overlay.data	Overlay group
overlay.text	
overlay.rdata	
main.bss	Bss group
overlay.bss	
overlay.sbss	New groups
overlay.sdata	
main.sbss	
main.sdata	GP →
main.data	Text group
main.text	
main.rdata	
system	

Changes to the Address file

‣ More overlay areas require more address grabbing

- Two loading addresses
  - One for the sdata file
  - One for the text file
- Others may required, depending on method used to clear bss areas

Changes to the Address file

‣ Example:

```
; load address for overlay sdata file
o1sdata    group
           xdef        SDLoadAddress
           section     .rdata
SDLoadAddress     dw          group(o1sdata)
; load address for overlay text file
o1text     group
           xdef        TextLoadAddress
           section     .rdata
TextLoadAddress   dw          group(o1text)
```

Other Considerations

‣ File loading

‣ Overlay bss areas

‣ Version of *psylink.exe* used

File Loading

‣ Two for each overlay
  - Plan ahead for loads
‣ Reading may be more problematic
  - High-level functions not suitable

Overlay BSS Areas

‣ These areas MUST be cleared
- Assembler routine
- Libc function

‣ Easier and faster if .sbss and .bss sections are next to each other

Overlay BSS areas

‣ Clearing with assembler routine

- Code is in *SNMAIN.S*, cut and paste in pertinent group/section names
- Call as a function

Overlay BSS areas

‣ Clearing with libc functions

 • Use *memset()* or *bzero()*

 • Get needed parameters from address file

 • These modules are tiny--use libc2 and avoid the speed hit from libc

Version of Psylink.exe

▸ Versions prior to 2.58 may produce .BIN files that are not the correct size

▸ If a group was placed in a file, and the following group was not, the following group would be included in the previous group's file