Executables & Arrays

CSE 351 Winter 2024

Instructor:
Justin Hsia

Teaching Assistants:
Adithi Raghavan
Aman Mohammed
Connie Chen
Eyoel Gebre
Jiawei Huang
Malak Zaki
Naama Amiel
Nathan Khuat
Nikolas McNamee
Pedro Amarante
Will Robertson

Relevant Course Information

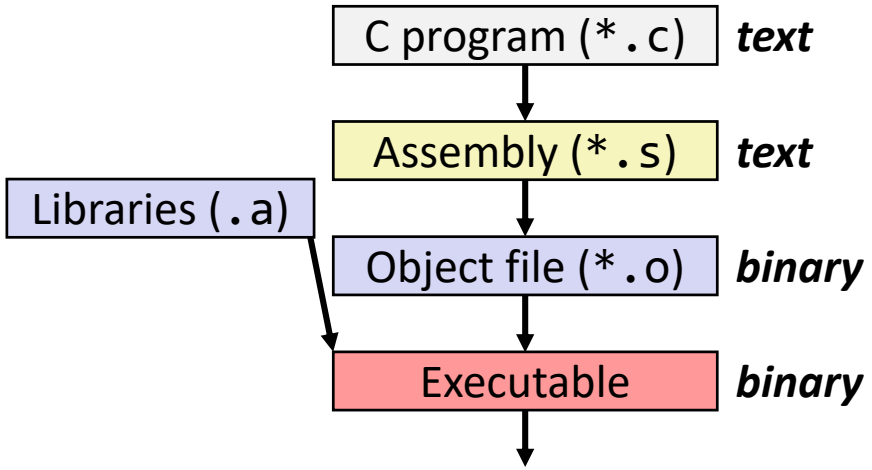
- Lab 2 & HW11 due Friday (2/2)
- HW12 due Monday (2/5)

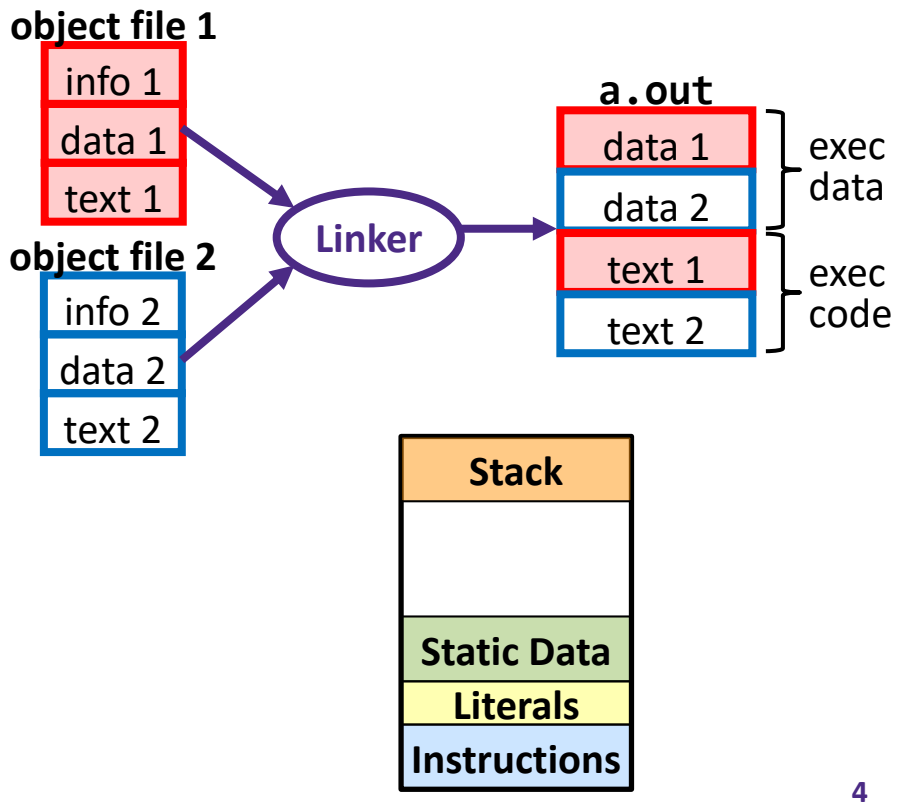
- Midterm (take home, 2/8-10)
  - Make notes and use the midterm reference sheet
  - Form study groups and look at past exams!
  - Mix of computational questions and open-ended short answer questions
  - Midterm review problems in section next week
  - Individual, but can discuss via “Gilligan’s Island Rule”

Executables & Arrays

Lesson Summary (1/2)

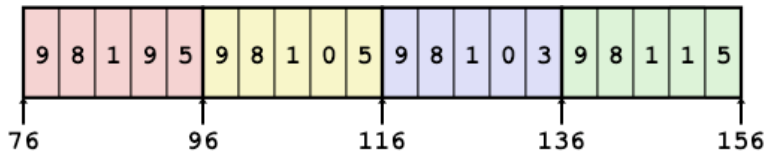
- Building an executable
  - Compiling uses specified optimizations to generate assembly code
  - Assembling produces object code in object files
  - Linking stiches together executable (machine code) using symbol and relocation tables to produce finalized addresses
- Loader sets up initial memory from executable
- A disassembler read object or machine code and tries to interpret the bytes as assembly

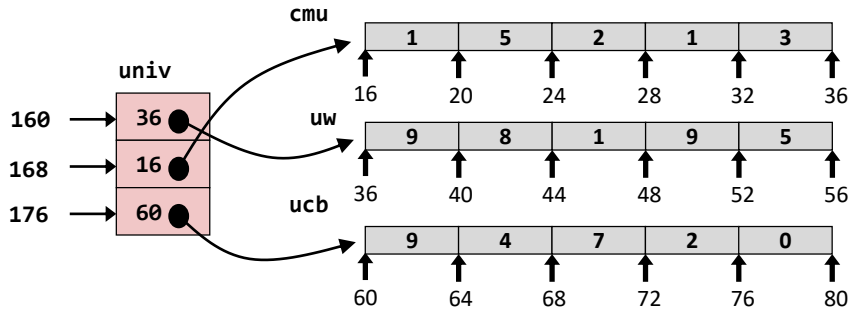




Lesson Summary (2/2)

- Arrays
  - One array declaration = one contiguous chunk of memory
  - No bounds checking (and no default initialization)
    - Accessed in assembly via `(Rb,Ri,S)` or `D(,Ri,S)`
  - *Array names* are not variables, but expressions that return the address of the array
    - Passing an array to a procedure really passes a pointer
  - **Multidimensional arrays**
    - Array of arrays in one contiguous block
    - Mem[`sea+20*r+4*c`]
  - **Multilevel arrays**
    - Array of pointers to separate arrays
    - Mem[Mem[`univ+8*r`]+`4*c`]





Lesson Q&A

- Learning Objectives:
  - Describe the key components of the CALL process.
  - Use `gcc` and `objdump` to extract information from each phase of CALL.
  - Analyze the memory allocations and accesses for arrays.

- What lingering questions do you have from the lesson?
  - Chat with your neighbors about the lesson for a few minutes to come up with questions

Executables & Arrays – Practice

Polling Questions (1/2)

- Use the following disassembly:

```
0000000000401126 <main>:
  401126:   48 83 ec 08        sub    $0x8,%rsp
  40112a:   bf 10 20 40 00     mov    $0x402010,%edi
  40112f:   e8 fc fe ff ff     callq  401030 <puts@plt>
  401134:   b8 00 00 00 00     mov    $0x0,%eax
  401139:   48 83 c4 08        add    $0x8,%rsp
  40113d:   c3                 retq
  40113e:   66 90              xchg   %ax,%ax
```

  - What is the byte of data at address **0x40113b**?

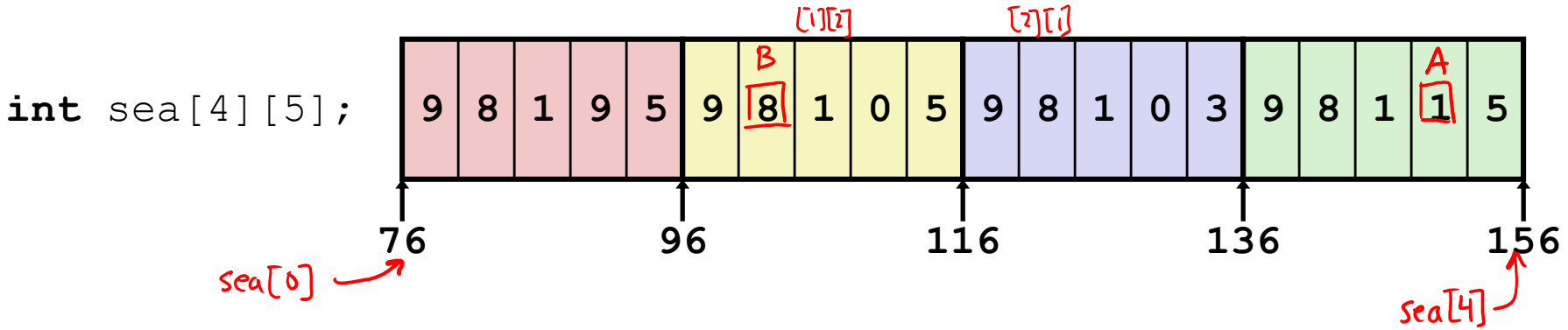
    0x c4

  - The immediate $0x402010 can be found in the machine code! **What is its address?**

    0x 401126

Polling Questions (2/2)

- Which of the following statements is FALSE?



A. `sea[4][-2]` is a *valid* array reference — Yes, returns 1

B. `sea[1][1]` makes *two* memory accesses — No, only single memory access

C. `sea[2][1]` will *always* be a higher address than `sea[1][2]` — Yes, because C is row-major

D. `sea[2]` is calculated using *only* `lea` — Yes, sea[2] returns address of array row

Executables & Arrays – Context

Mid-Quarter Survey

- No context today!

  Please take this time to fill out the mid-quarter survey:
  https://canvas.uw.edu/courses/1695950/quizzes/1956436

Group Work Time

- During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the homework problems
  3) Work on the lab (if applicable)

- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support