



King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Computer Engineering Department
COE 301: Computer Architecture

LAB 05: Arrays and Files

Saleh AlSaleh



Agenda

- Static Allocation: Declaration and Initialization
- Dynamic Allocation using System Call
- Address Calculation of 1-D and 2-D arrays
- Files: Open, Read, Write, Close
- Live Examples
- Tasks

Static Allocation

- Allocates one variable or an array of variables in the static area of data segment.
- Array size is determined at assemble time.
- Data Types: byte (1 Byte), half-word (2 Bytes), word (4 Bytes)
- Declaration and Initialization at the same time Example

```
.data  
secretbyte: .byte 0xAB  
secrethalf: .half 0xABCD  
secretword: .word 0x89ABCDEF
```

```
.data  
secretbytes: .byte 0xAB:10  
secrethalves: .half 0:15  
secretwords: .word 1:20
```

- Declaration only Example

```
.data  
secretarr: .space 100
```

Dynamic Allocation

- Allocates one or more bytes at run time in the dynamic area (heap) of the data segment.
- Some programs require different array sizes based on some inputs.
- Use system call (**\$v0** = 9) and number of bytes to allocate in **\$a0**.
- The base address will be returned in **\$v0**, this base address needs to be saved.

Address Calculation

- 1-D Array Address Calculation

- arr1D: .type 0:20
- For example int arr1D[20];
- Address of arr1D[i] =
$$\text{base_address}(\text{arr1D}) + (i * \text{element_size})$$

- 2-D Array Address Calculation

- arr2D: .type 0:20
- For example int arr2D[4][5]
- Address of arr2D[i][j] =
$$\text{base_address}(\text{arr2D}) + (i * \text{col_size} * \text{element_size}) + (j * \text{element_size})$$

Files

- Files are very useful for many applications.
- They ease the input and output of values needed by the program.
- In order for a file to be used, it needs to be opened first.
- System Call 13 is used to open a file with the following options:
 - **\$a0** address of null-terminated string containing the file name.
 - Path can be relative to the location of MARS.jar file or an absolute path.
 - **\$a1** = 0 for read-only.
 - **\$a1** = 1 for write-only with truncate and create.
 - **\$a1** = 9 for write-only with create and append.
- It returns in **\$v0** a positive file descriptor if it can open the file or negative if error.
- File descriptor needs to be saved for other system calls.

Files (continued)

- System Call 14 is used to read file contents with the following options
 - **\$a0** file descriptor
 - **\$a1** = address of input buffer
 - **\$a2** = maximum number of characters to read
- It returns in **\$v0** positive number of characters read, zero if end of file or negative if error
- System Call 15 is used to write contents to file with the following options:
 - **\$a0** = file descriptor
 - **\$a1** = address of output buffer
 - **\$a2** = number of characters to write
- It returns in **\$v0** positive number of characters written or negative if error
- System Call 16 is used to close file with **\$a0** containing the file descriptor.

Live Examples