SE2031 Software Tools - Structures and Unions Summer 2010 CSE2031 Software Tools - Structures and Unions Summer 2010 Przemysław Pawluk

Department of Computer Science and Engineering York University Toronto

June 1, 2010

zemysla Pawluk ructures Continu ion inked List inked list nd Arrays rees nions t-fields

rzemysi Pawluk tructure Contin ion inked Li inked lis nd Array

rees nions

VORK

What have we done last time?

• Memory Allocation

Structures

Table of contents

s	Structures – Continuation
~	 Linked Lists
	Linked list and Arrays

Trees

2 Unions

3 Bit-fields

4 Review

Notes

Notes

Linked List

CSE2031 Software Tools -Structures Ind Union Przemyslaw Przemyslaw Pawluk Structures - Continu-ation

Notes

Widely used structure i.e. to implement queues.

- Has head and tail
- head is a list element
- tail is a list
- $\bullet\,$ each element points to the next one
- last element points to NULL

Software Tools -tructure ad Union rzemysl: Pawluk tructure Continu-ion inked List inked List inked List inked List inked sist nions it-fields

FIFO queue

- elements are added at the end
- ${\ensuremath{\, \bullet }}$ elements are taken from the beginning





Notes

Hash-table – Table lookup

Notes

Hash-table in this case contains pointers to linked lists

- Flexible structure used to store multiple elements (i.e. text)
- Improves search
- Two methods are provided:
 - install(s,t) adds element s and replacement text t
 lookup(s) looks for s in our hash-table

U

SE2031 Software Tools -tructures Id Union rzemysla Pawluk tructures Continu tion inked List inked list inked list on darao



Notes

Notes

How to calculate hash? • Hash is small number (between 0 and HASH_SIZE) • Hash is calculated by hash function • Hash is calculated based on value Example unsigned hash(char *s){ unsigned hashval; for (hashval=0; s!=' 0'; s++) hashval=*s + 31 * hashval; return hashval % HASHSIZE; }

Binary Tree - dictionary

- Tree has exactly one root element
- Each element has at most two children
- Each element stores a word and its translation
- Left child is less-or-equal than parent
- Right child is grater than parent



		Good	to	know	about	structure
--	--	------	----	------	-------	-----------

Notes

Notes

- structure may contain virtually instances of any type but ...
- structure cannot contain an instance of itself,
- structure can contain a pointer to itself,
- the size of structure is **not** necessarily equal to the sum of sizes of members (depends on implementation)

Union

- Variable that may hold at different times objects of different types and sizes
- Compiler keeps track of size and alignment requirements
- It's programmer's responsibility to keep track of which type is currently stored in a union
- type retrieved must be the one most recently stored
- implementation-dependent results if something is stored as one type and retrieved as another
- access to the union members is syntactically the same as to structure members union.member or union_ptr->member
- union is large enough to store "widest" member
- all members are stored in the same are in memory

Tools ructu d Uni

tructure Continu on ked Lists ed list Arrays

Initialization

Union may only be initialized with a value of the type of its first member! $% \label{eq:constraint}$

15/3

I	Union init example
	Correct
1	union {
2	int x;
3	float y;
4	char *sptr;
5	u = 1;
	Incorrect
1	abov a [] "teat".
1	char s[] = test;
2	union {
3	int x;
4	float y;
5	char *sptr;
6	} u = &s
	1 2 3 4 5 1 2 3 4 5 6

	Bit-fields
CSE2031 Software Tools - Structures and Unions Przemysław	 Used when storage space is at premium Allows to pack several objects into a single machine word Can be used to implement flags or masks
Pawluk	Masks using define
Structures - Continu- ation 1 Linked Lists 2 Linked list and Arrays Trees 3 Jnions	#define KEYWORD 01 /*0000 0001*/ #define EXTERNAL 02 /*0000 0010*/ #define STATIC 04 /*0000 0100*/
Bit-fields	Maalua wainan hit fialaa
Review	Wasks using dit-fields
1 2 3 4	<pre>struct{/*one bit per flag*/ unsigned int is_keyword : 1; unsigned int is_extern : 1; unsigned int is_static : 1;</pre>
18/30 5	} flags;

Notes

Notes

C-basics

ructure Contini on 1ked List 1 ked list d Array

- Program structure
- variables' types
- mixed type arithmetic
- cast
- precedence of operators
- conditional expressions
- pre- vs. post-
- $\bullet\,$ numbers in C

20 / 30



Preprocessor

• #declare

- #include
- #if, #elif, #else and #endif
- defined(name), #ifdef and #ifndef

Notes

Notes

Notes

ructure Continu inked List inked list nd Arrays rees nions it-fields

Scope

• Declaration vs. definition

external vs. internalstatic and extern

scope

Tools -Structures and Union Przemyslau Pawluk Structures - Continuation Linked Lists Linked list and Arays Trees Unions Bit-fields

23 / 30

ructure Contine on sked Lis sked list d Arrays

Testing

Notes

Notes

- Random tests
- Black-box tests
- Glass-box tests
- Regression tests
- Boundary conditions testing
- Pre- and Post-condition testing
- Assertions

Strottes Strot

Pointers

YOR

CSE2031 Software Tools -Structure nd Unior 'rzemysla Pawluk tructures Continu tion .inked List ind Arrays Frees Inions

Definition

- Access to a pointee
- Getting the address of a variable
- Arithmetic
- Pointers and Arrays
- void*

26 / 30

ORE

rzemysla Pawluk tructures Continuinked List inked list nd Arrays 'rees nions

Different definitions

Notes

Notes

- to simple types
- to arrays, structures and unions
- to functions

Memory allocation

- malloc
- calloc
- realloc
- free

Notes

28/30

YORK

tructure Continu ion inked List inked list nd Arrays

Structures and Unions

- Structures
 - DefinitionLinked Lists (FIFO, stack)

 - TreesHash tablesbit-fields

CSE2031 Software Tools -Structure and Union Przemysla Pawluk structures Continu-tion Linked Lists inked list and Arrays Trees Inions

- Definition
- Properties
- initialization

• Unions

- namespace
- deep vs. shallow copy

rzemyslar Pawluk tructures Continu-tion inked lists inked list nd Arrays Trees nions it-fields

Lab-test

Notes

Notes

- 3 programming tasks
- $\bullet~\text{standard}~\text{I/O}$
 - arrays, structures and unions
 - memory allocation