turday
13-06-2020

CD Assignment-4

UNIT-4

N.V.S.k.kalyani
174N1A0584
3rd B.tech CSE-C

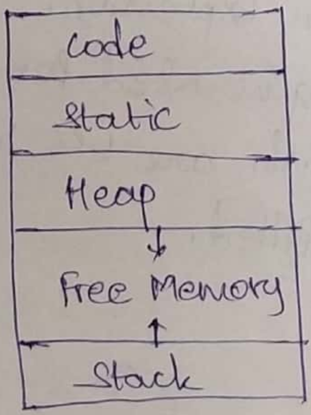
2 Marks

1A) Running Storage:-

The memory space required for executing the program is called as "run-time storage". This runti storage must fold the following:

- The generated target code
- Data objects
- control stack to keep track of procedure activations.

The sub-division of run-time storage is shown below:



2A) In variables length data format, memory space is allocated for a variable depending on the valu assined to it.

Ex:- In PL/SQL the variable length data can be declared as var char a[10] = "Hello"; Here stora is allocated for 5 bytes.

3A) Symbol Tables stores the following Info about iden

- The name
- The data type
- Size
- Line of Declaration
- Line of usage
- Address

4A) Symbol Table is a data structure used by compiler to store information, about source program contracts while the program is being compiled.

usually, Basic operations on symbol-table Include the following:

1. Insert a new symbol into the symbol table.
2. Remove a symbol from symbol table.
3. lookup- to search for a name & return a pointer to its entry.
4. Free - to remove all entries and free the storage of the symbol table.

5A) Stack allocation limitations:-

* The size of the data object and contraints on its position in memory must be known at compile time.
* Recursive procedures are restricted, because all activations of a procedure use the same binding for local names.
* Data structures can't be created dynamically, since there is no mechanism for storage allocation at run time.

10 Marks

1A) **Activation Record:**

Information needed by a single execution of function is managed using a continous block o storage called as "activation Record".

- Each live function has a activation method.
- The current execution path specified by stack represents the recently activated function activa-tion record in the top of the stack.

Activation Record Format:-

Actual parameters
Returned values
control link
Access link
saved machine status
local Data
Temporaries

Temporaries:- Temporary variables used during evaluation of expressions.

local Data:- It is the data that is local to the execution of procedure.

saved machine status:- This field holds informa-tion regarding the status of machine just before the procedural is called this field contains the machine registers and pc values.

Access link :- It refers to the non-local data in other activation records that is needed for the current procedure call activation. It is also called as static link.

control link :- It points to activation record of calling procedure. This link is also called an dynamic link it is optional field.

Return value :- stores return value of functional call.

1B) Heap managment mechanism :-

Heap Area holds all dynamically allocated variables information. The heap is the portion of the storage that is used for data variables that lives indefinitely or allocated and de-allocated dynamically.

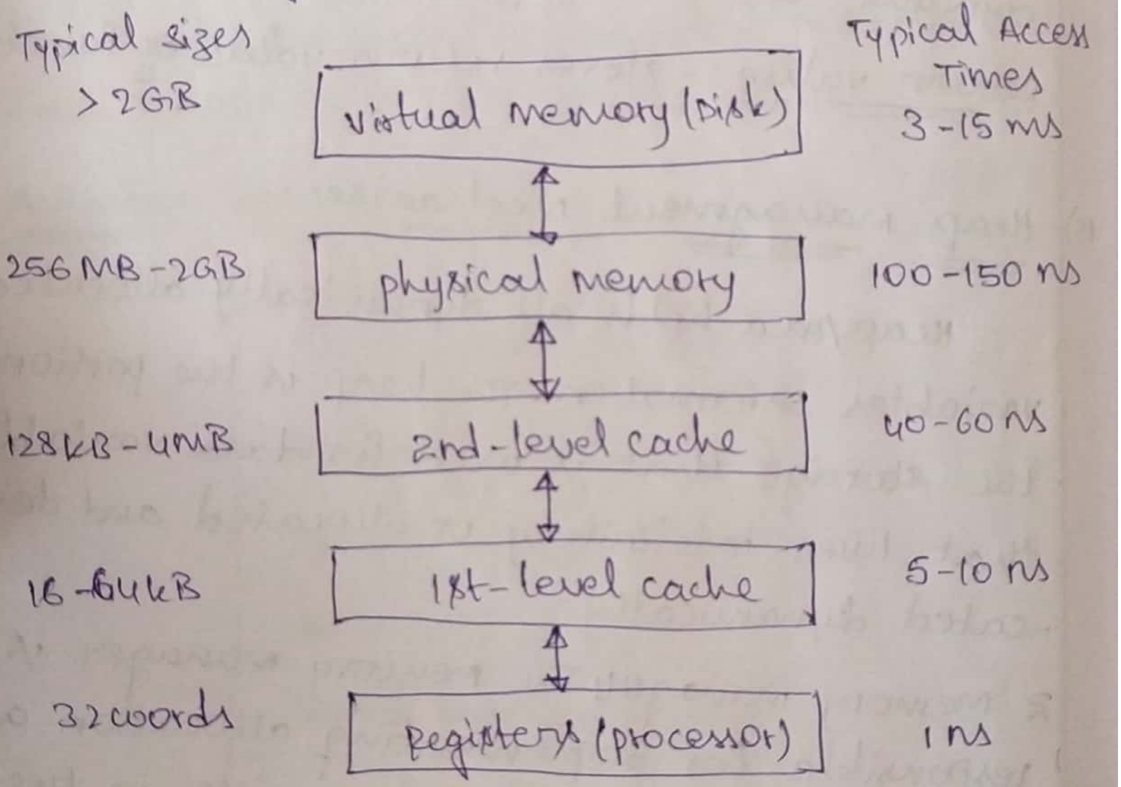
2. Memory Manager: The memory manager is responsible for implementing allocation and deallocation strategies space with in the heap.

It serves as an interface between application program and the operating system. Two basic functions are : i, Allocation
ii, De-allocation

Desired properties of memory manages include the following a. Space efficiency
b. program efficiency
c. low overhead.

3. Memory Hierarchy of a computer : particularly all modern computers arrange their storage as a memory hierarchy.

A memory Hierarchy, as shown below consists of a series of storage elements with the smaller faster ones "closer" to the processor, and the larger slower ones further array.



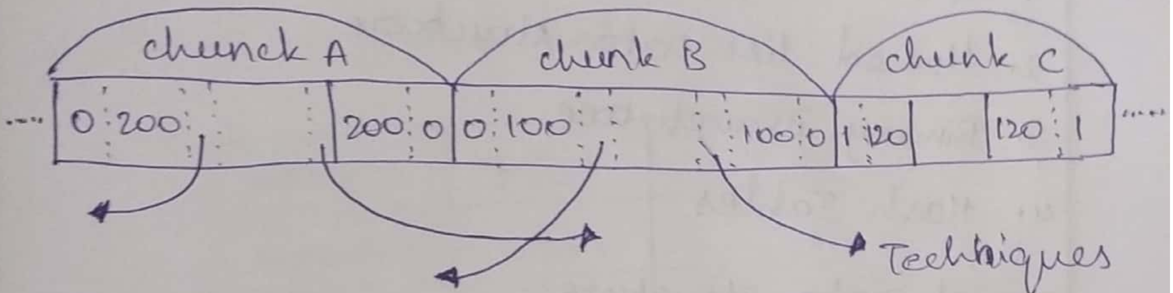
4. locality in programs :-

* most programs exhibit a high degree of locality that is they spend most of their time executing a relatively small fraction of the code and touching only small fraction of the data.

* Temporal locality

* Spatial locality

5. Reducing Fragmentation :- As the program allocated and deallocates memory. This space is broken up into free and used chunks of memory, and the free chunks need not reside in a contigous area of the heap. we refers to the free chuncks of memory as holes.



To overcome that we use the following

* The Best-Fit and Next-Fit object placement
* managing and cookescing Free space.

2A) Symbol Table organization for Block-structured language :-

Introduction :-

A programming language that permits the creation of block within a program is known as a Black-struct used language.

It is a class of high-level languages in which a program is made up of block. A block may Include nested blocks as components. such nesting being repeated to any depth. usually a blocks consists of a sequence of statements and/or blocks, preceded by declarations of variables.

Ex: C, C++ and Java languages divides program statements into blocks like functions, loops etc delimited by braces { }.

Following are commonly used data structure used for symbol table organization in Block structured language.

1. List Data structure
2. Linked list Data structure
3. Binary Branch tree
4. Hash Tables

1. List Data structure:-

Linear list is the simplest kind of mechanism to implement symbol table.

In this method, array is used to store name and associated. New names can be added in the order they arrive. The pointer "available" is main-tained at the end of all stored records to indicate empty slots. The symbol Table maintai by list data structure is shown below.

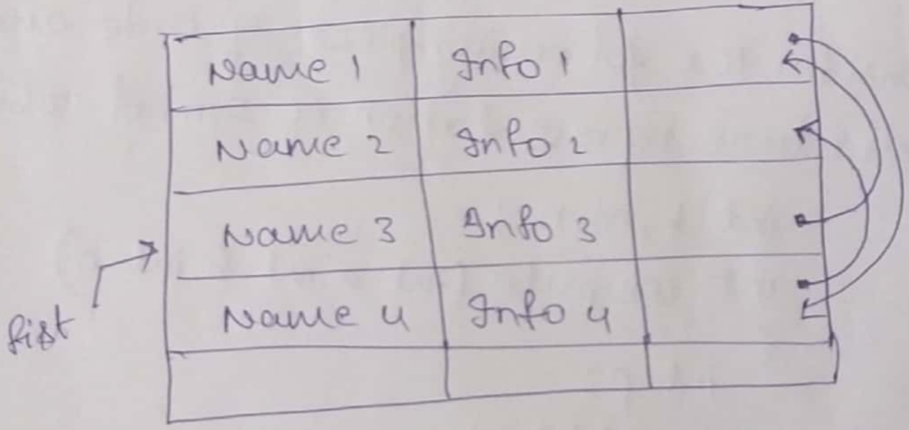
Name 1	Information1
Name 2	Info 2
⋮	⋮
Name n	Info n

Available
start of empty slot

2. linked list Data structure:-

This symbol Table representation uses linked list. A link field is added to each record. we search the records in the order pointed by the link field.

A pointer "first" is maintainined to point, to the first record of the symbol Table. we search the records in the order pointed by the link field. The format is as shown below:



Name 1	Info 1	
Name 2	Info 2	
Name 3	Info 3	
Name 4	Info 4	

3. Binary search Tree:

The symbol table is represented as a binary tree format, where the code structure is follows:

left child	symbol Name	Information	Right child

The left child stores address of previous symbol and right child stores address of next symbol.

4. Hash tables:-

In hashing scheme, two tables are maintained a hash table and a symbol table.

* The hash table consists of k entries from 0 to k-1. These entries are basically pointess to the

Symbol table pointing to the names in the symb table.

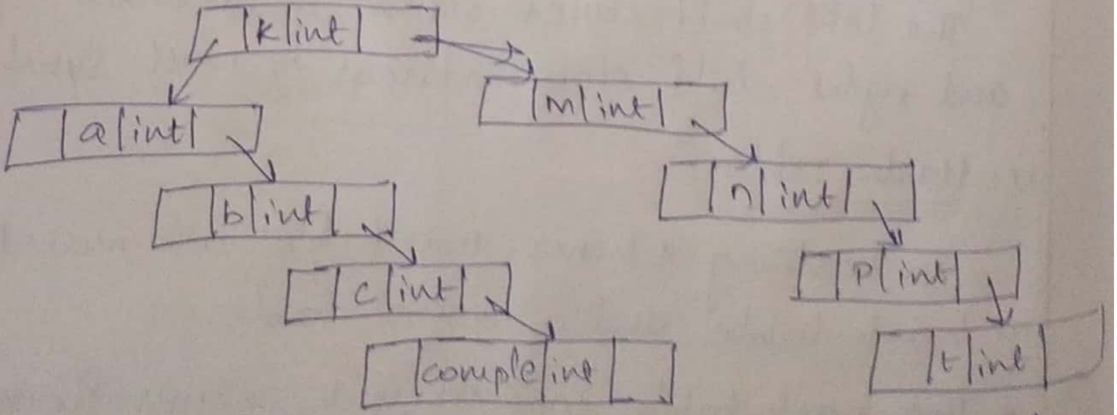
* To determine wheather the name is there in the Symbol table. we use a hash function "h" such that h(name) will result any Integerx b/w 0 to k-

we can search any name using the function as shown below

position = h(name);

Ex:- consider the following piece of code and crea Tree structure representation of symbol Table

```
int k,m,n;
int compute (int a,int b,int c)
{
   int p;
   p = a+b+c;
   return (p);
}
void main()
{
   int t;
   t = computed (10,20,30);
}
```



It's Binary tree representation of symbol table.