

ADVANCED OPERATING SYSTEM

ASSIGNMENT -2

MCA – IV SEMESTER

DISTRIBUTED LINEAR-SEARCH

SUBMITTED TO:

Dr. Sapna Varshney

SUBMITTED BY:

Priya Jain(22)

Radha Verma (24)

Problem Statement 2 - Delegated Linear Search

Objective: Implementation of Distributed Linear Search using Linux System calls - An array is split up into sub-arrays and searching the sub-arrays for an integer is delegated to other processes.

Input: The program should take two arguments as input - an array of integers to be read from a file (first argument is the path to the file) and an integer to be searched (second argument).

Output: Print index of the number to be searched if found, or 'Number not found'.

Approach:

The main program reads the complete input array as a global array. The main program creates 2 processes and gives one half of the input array to the 2 child processes to search the required number. If the size of the segment is small enough (say ≤ 5), then it searches the number in the segment by itself. Otherwise, the task is delegated by creating more child processes wherein each process is given a segment of the array to handle and the number to find.

If any process finds the required number, then it sends a signal to the main process and also returns the index. After this, the main process kills all the other processes and prints the index. The code should also handle the case when the number is not present in the array.

PIPE is chosen as the method of IPC mechanism.

Functions Used:

`int readfile (char*file):` Returns the count/size of the file and manipulates a global array of 1000 elements to store the integers stored in file.

`int search (int array[], int left, int right):` performs the linear search on the given array. It is called when the array size is less than 5. It returns the position of the element in sub-array +1 to indicate the actual position of element (as array starts with 0th index).

`int linear_search(int array[], int left, int right):` It checks the size of array is less than 5. It calls for search to perform basic linear search. if not the process will fork () and create a child1(left child) to handle the left part of the array recursively.

Similarly, it will fork () again to create a right child to handle the right sub-part of array.

If found in any of the process forked above the process will return the value found while writing in the pipe created.

`int main():` Main process will handle the main mechanism of the program .As mentioned in problem statement two child will be created of the main process to handle the original array by dividing it into sub-arrays. Pipe reader process will remain close while writing and writer process will remain close while reading.

Illustration:

Let the array be of size 32. Initially the two child process will handle sub-arrays.

Left child -> 0-16

Right child->17-32

After recursive call the arrays will be divided in subsequent halves. The call to each process is based on context switch.

