

CSCI 3020U Lab 4
Host Dispatcher Shell Project
Design Document

Kevin Lovell (100559665)
Mitchell Childerhose (100553756)
March 27th, 2017

1. a

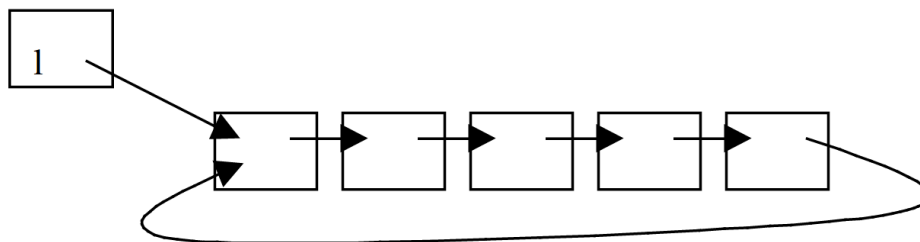
To solve the issue of memory storage allocation we could use one of several algorithms. First-fit would allocate memory to the first available hole in memory. Best-fit is a another algorithm that would allocate memory to the smallest available hole that can fit the given data, but leave the smallest possible leftover holes. This requires searching the entire memory block every time new data is stored. In contrast Worst-fit is an algorithm that would allocate memory to the largest available hole and produce the largest leftover holes for more data. This would also require searching the entire block when new data is stored. For this lab we will use the first-fit algorithm for its simplicity and its faster search times.

1. b

This dispatcher uses the Round Robin approach to dispatching and queueing. Each process is assigned a fixed time by the scheduler to be completed. The process gets terminated if it completes within the time frame it is allocated. If the process does no complete within the time frame, it is rescheduled to be executed after all the other processes are given a chance to run. Round robin makes use of the Shortest Job First(SJF) and First In First Out(FIFO) paradigms for scheduling. No priority is given to each process, therefore, starvation cannot occur. The process arrival time dictate the order that the time units are allocated in.



The data structure used to implement the round robin algorithm is a simple circular linked list. Each node represent and process and points to the next node. This allows the dispatcher to cycle through each of the process and give each of them a slice of time to execute. This is convenient for the OS as it can loop back the beginning of the linked list (or queue) after each process has been given its time to run.



The dispatcher will start at the beginning of the linked list of processes, and utilize the First Fit algorithm by getting the first memory space that the process will fit into and store it. The dispatcher will then move onto the next node(process) and look for the first fit of memory until the end of the list.

1. c

hostd.h: Includes relevant data structures

hostd.c: Contains main

utility.c: Uses best fit to locate and allocate memory where space is available. Loads dispatchers.

Utility.h: Contains structs for resources and process details. Allocates continuous chunk of memory to resource struct. Frees allocated chunk of memory. Parses the file.

queue.h: Contains linked list for queue and declares push pop functions

queue.c: Contains FIFO queue functions push and pop

sigtrap.c: Ticks away at reporting process and reports symbols. Contains code to display different colour combinations to ASCC terminal.

The program loads the dispatch list from a text file in the format:

```
<arrival time>, <priority>, <processor time>, <Mbytes>, <#printers>,    <#scanners>,  
<#modems>, <#Cds>
```

It contains comma separated values, for example:

```
12, 0, 1, 64, 0, 0, 0, 0  
12, 1, 2, 128, 1, 0, 0, 1  
13, 3, 6, 128, 1, 0, 1, 2
```

It adds each process instance to the job dispatch list queue, iterates through each item in the list, and adds each process to the appropriate queue. Then it allocates the resources available to each process and executes them in sequential order using the fork() and exec() commands. The resources are allocated using the First Fit algorithm assign memory space to each process. It iterates through the linked list until all the processes have been fully executed and the queue is empty.

1. d

A multilevel dispatching scheme is needed to allocate memory efficiently and effectively as well to schedule processes in a reasonable and fair system. The many levels are needed to assign different levels of priority to processes as they appear in the queue.

There are a few shortcomings of the Round Robin dispatcher. One issue is that it does not give higher priority to more important processes due to its linear nature. Urgent requests cannot get handled faster than least important requests in the queue. Real operating systems that use the Round Robin technique include a tool to suspend running processes to give priority to a “rush job”. Wasted time and lost resources can result from the cascading completion of the processes.

Another shortcoming occurs in memory allocation when using the best fit algorithm. Although it is the fastest allocation strategy, it leaves over wasted memory with blocks too small to fill and can't accept requests for larger memory requirements down the line. This could be resolved in a real operating system by using fragmentation. This would break memory down to fixed block sizes and avoid having smaller holes randomly throughout memory.