Page 12

Page 1

Set the parameters to variable partition with no compaction. Try

the different placement policies out. Which seems superior?

**Example 3**

Enter four processes each 50k long, with an arrival time of 0 and runtime of 5.

Now set the parameters to variable partition, with no compac- tion and first fit placement. Run the simulation and notice the results. Now amend the parameters and set to variable partition with partition sizes of 300k and 600k. Run the simulation again.

Which memory management system seems to be the superior,

and why?

Now clear the process list completely and re-enter as follows:

length

400 350 150

50 50 50

arrival time

000111

runtime

882777

Now run the simulation with variable partition, no compaction

and first fit placement. Note the result. Now change to fixed

partition with partitions at: 500, 850, 900 and 950. Do not

change any of the other parameters. Run the simulation again.

Which seems to be the superior method now? What can be said about average/maximum process length and its bearing on our

choice of memory partition?

Page 2

Table of Contents

Overview ............................................................3

Theory ................................................................3

Page 11

You can terminate the simulation at any point by clicking on the Exit button. This immediately stops the simulation run, bypasses the results screen, and returns you to the main menu so that you can adjust the process list or parameters.

**Example Process Data**

**Example 1**

Enter the following process list in the exact order below:

Using the Simulation program............................8

Update Process List Screen ................................8

Amend Parameter Screen ...................................9

length

200 300 200 500

arrival time

0006

runtime

10

5

10

5

The Simulation Screen .....................................10

Example Process Data ......................................11

To use this program, you will require a PC with a 32 bit operating system, such as Windows 95, 98 or NT. The program file is called OSsim.exe and you should double click on this filename in Windows Explorer to start the program.

Set the parameters to variable partition with no compaction and

with a first fit placement policy. Now run the simulation. Notice how process D fails to be allocated to memory due to fragmen- tation. Now try the simulation run again, with the same process list, but with the various compaction policies. Notice the effect of each. Try altering the placement policy with compaction on. Note the effect.

**Example 2**

Process list:

length

200 200 150 100 300

arrival time

00057

runtime

10

3

10

54

Page 10

Page 3

partitions, then you must enter at least one partition in the list.

Remember that these are not the partition sizes, but the partition locations in memory.

**Overview**

This program simulates how an operating system copes with

e.g.

200, 400, 600

memory allocation in a multi-tasking environment. Normally, processes start at random intervals and are of a random length.

Would specify that there were four partitions. The first of 100k

length would run from 100 to 200k. The second of 200k would

run from 200k to 400k. The third of 200k would run from 400 to 600k. The forth of 400k would run from 600 to 1000k. Remem-

ber that the operating system itself occupies the first 100k of memory and the top of memory is 1000k.

Each partition length must be a multiple of 50k.

The 'halt simulation after each event' tick-box will pause the

simulation when something interesting happens. Once you have finished amending the parameters, use the main menu button to return.

**The Simulation Screen**

Selecting this button from the main menu will start the simula- tion and execute the process list using your chosen parameters. The clock will commence with the earliest process in your list, which is process A. Once the process list is exhausted and any

processes in memory have used up their runtime, the results

screen will be posted. You can print the listing to your Windows default printer by clicking on the print button. Use the Exit but- ton to return to the main menu.

If you selected the appropriate tick-box on the amend parame-

ters screen, then the simulation run will halt after each event. Alternatively, you can halt the simulation run at any time by

clicking on the pause button. Click the button again to resume execution.

However, this simulation program allows you to set up your

own process list and decide how long processes are, when they

arrive and how long they run for. In this way, you can use the

same process list and test to see how each of the various policies

cope with memory allocation. You should be able to work out which methods are superior and under what circumstances.

Before explaining how the simulation works, let us look at the various problems faced by an operating system with regard to memory management.

**Theory**

In a single task environment, only one process is run at any one time. As long as the process will fit within the memory available

then it can be executed. When the operating system is set-up to

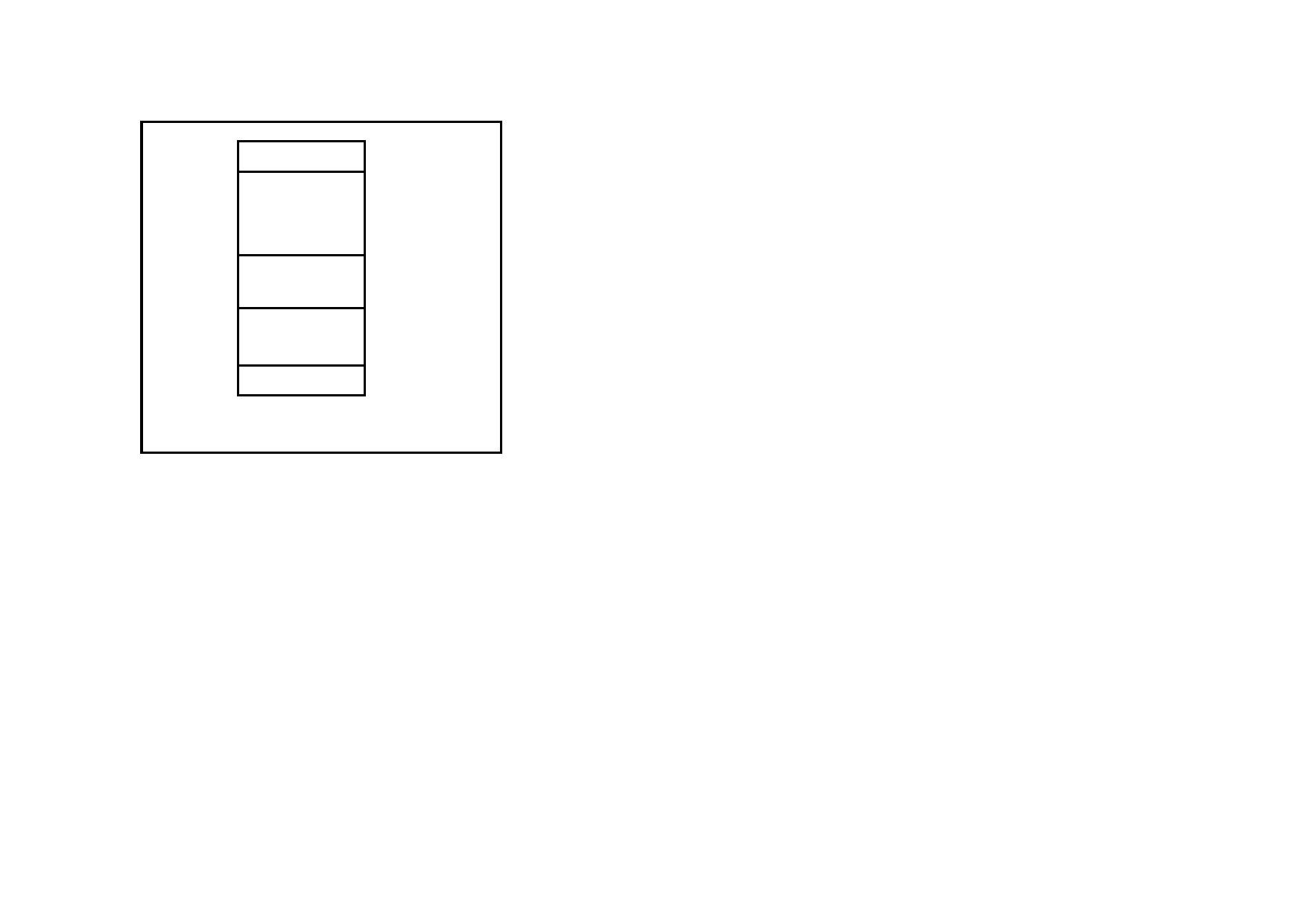
multi-task, then the memory has to be split up and allocated between the processes in some way.

One solution to this problem is to divide the memory up into sections called fixed partitions. Figure one shows just such a situation. The total memory available runs from 0k to 1000k.

The first 100k is taken up by the operating system. The remain-

ing memory has been sectioned into two partitions, one 400k long and the other 500k long. Two processes can now be allo- cated to memory and as long as the processes are small enough

to fit into one of the partitions, then the operating system can accommodate them.

Page 4

(free)

1000k

multiply of 50k.

Page 9

Process B

(free)

Process A

Operating system

Figure one

800k

500k

300k

100k

2. Arrival time. This is the clock time at which the process ar-

rives. The smaller the arrival time, the earlier the process ar-

rives during the simulation run. The earliest clock time that you can specify is zero. Do not make the clock times too far apart or

during the simulation there will be long periods when nothing happens. Remember that during the simulation run, the clock increments about every second.

3. Runtime. This is the number of clock cycles that the process will remain active and therefore allocated to memory. Once the runtime has expired, the process has been successfully run and the memory will be freed for re-allocation.

The add button will allow you to add the process details listed

on the right into the process list. Only 26 processes are allowed

and are allocated the letters A to Z ,depending on their arrival

time. If you add two processes with the same start time, then the process you add first is deemed to be the earliest.

However, a problem immediately becomes apparent, because the

processes do not completely fill the partitions that they occupy.

Process A is only 200k long and so 200k of memory has been wast- ed within the partition. This is known as internal fragmentation. The

other partition has a similar problem as only 300k of the 500k has been used.

Altogether then, 400k of the available memory is unavailable for use and effectively wasted until the processes terminate and new ones allocated.

A solution to this problem is to have variable partitions. With this

system, the memory is divided up depending upon the length of the processes. Figure two shows how the fragmentation problem seems

to have been overcome, because the same two processes allocated to memory under this system frees up the 400k for use by another process.

The delete button will removed a process from the list. Simply

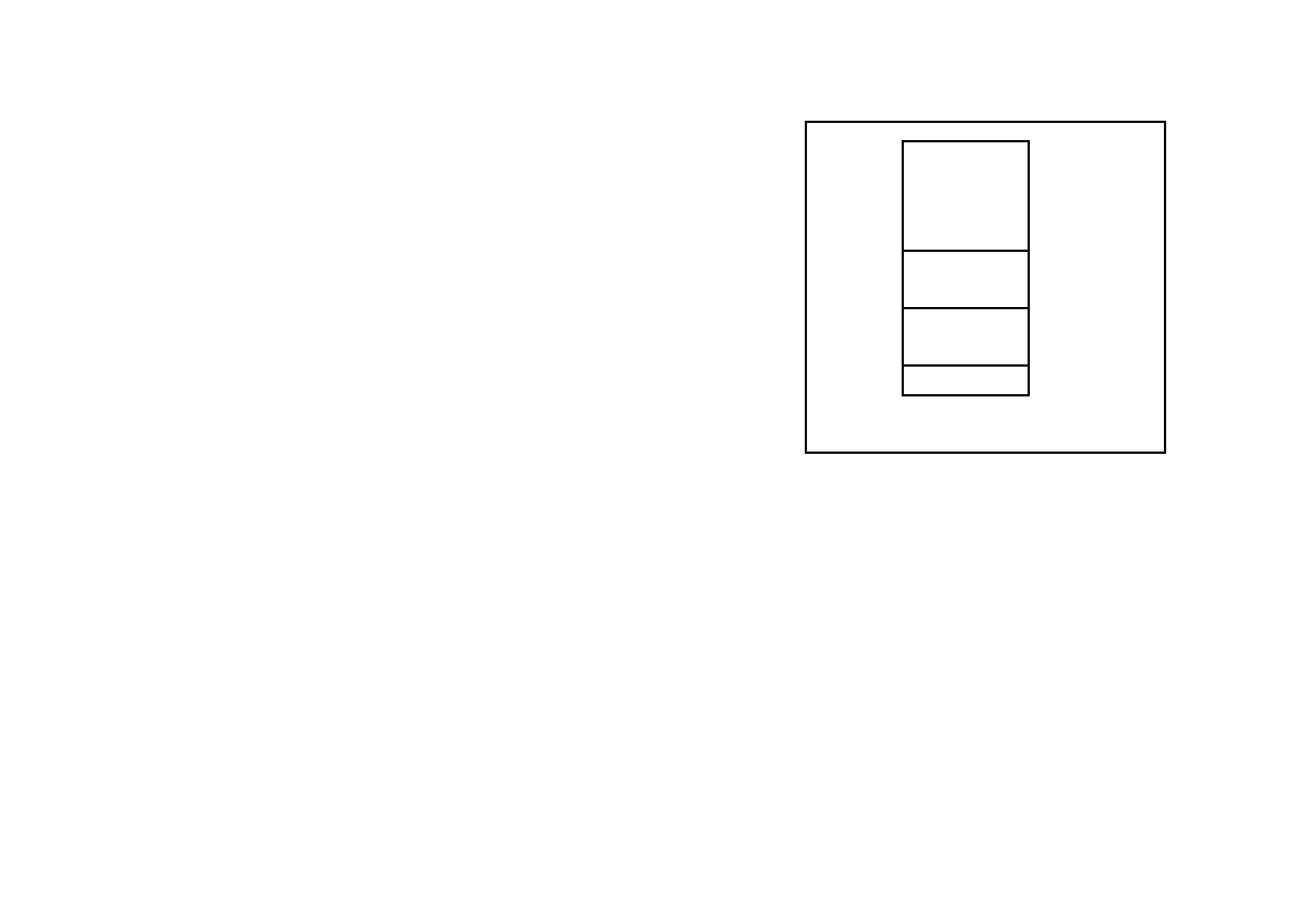
highlight the process that you wish to remove by clicking on the list item with the mouse and then select this button. The process list will automatically be renamed, if necessary, as the list must be contiguous.

The clear list button will remove all items from the process list.

Use the 'exit to main menu' button to return once you have fin- ished.

**Amend Parameters Screen**

Choose the options that you desire. Remember that compaction only works with variable partition memory. If you select fixed

Page 8

**Using the Simulation Program**

Upon starting the program, you will be presented with the splash screen and then the main menu. In order to run a simulation, you

should proceed as follows:

1. Set up a list of processes via the Update Process List menu op- tion. This is mandatory. (Lists of processes which produce inter-

esting results are contained in the example process data section of this guide).

2. Use the Amend Parameters menu option to select fixed/variable partitions, partition sizes, compaction and placement policies.

You can also elect to have the simulation halt after each event of significance.

3. Use the Start Simulation button to execute the process list. The

results will automatically be listed once the simulation has fin-

ished and can be printed out for reference.

(free)

Process B

Process A

Operating system

Figure two

1000k

600k

300k

100k

Page 5

4. Once the simulation has finished, go back to the Amend Param-

eters menu and try different parameters for the same process list to see how the different methods cope.

The individual menu options are now be described in more detail.

**Update Process List Screen**

This screen lists the current processes on the left. Individual pro-

cesses can be entered using the three boxes on the right and added

to the list. There are three parameters for each process:

1. Size. This can range from 50k to 900k. The process must be a

However, let's say that process C arrives, which is 150k long,

and that process B has now terminated. The resultant memory map is shown in figure three. It is now apparent that the frag-

mentation problem has raised it head again. This time there is a block of memory available for process allocation, but it is only 300k in size as it is sandwiched between two active processes.

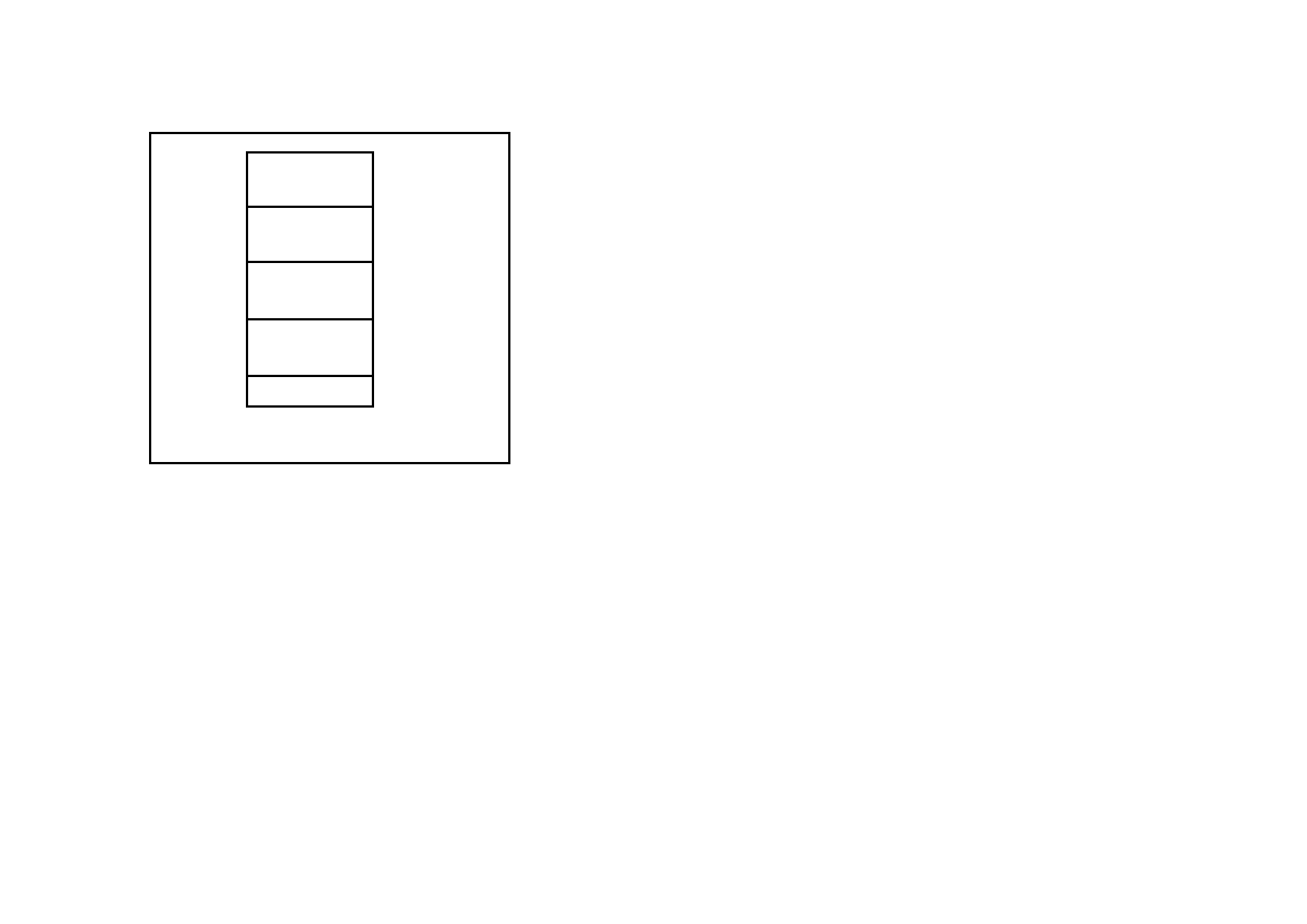
Unless a process starts which is exactly this size, then there will be some wastage. This is known as external fragmentation.

Another issue raised by fragmentation is that of placement. By referring to figure 3 again, it can be seen that if a new process

comes along which is 50k in size, there are two places to which it could be allocated; either at 300k or 750k. Putting the process

at 300k will leave a free memory block, or 'hole' of 250k and placing at 750k will leave a 'hole' of 200k.

It can be seen that a placement policy has to be arrived at. There

Page 6

Page 7

1000k

Another feature of variable partition allocation is that of coa-

lescing. In figure three, if process A terminated, there would be

two holes left next to each other. If these two were merged to-

(free)

Process C

(free)

Process A

Operating system

750k

600k

300k

100k

gether, then a single larger hole could be created, which would be advantageous for process allocation. Merging two holes to- gether is known as coalescing. In the simulation program, coa- lescing of holes is automatic.

Another way of improving the situation illustrated in figure

three would be to move process C down in memory to location

300k. This would leave a larger hole at the top of memory for process allocation. This process is called compaction (not de- fragmentation as you might expect!).

Additionally, it is necessary to decide at what stage to compact.

Two events trigger the compaction process:

Figure three 1. When a process terminates

2. When a process cannot load due to fragmentation.

are three available:

1. First fit—Place the process into the first 'hole' that it fits into,

starting the search at the lowest memory location. In the example, the process would be allocated starting at location 300k.

2. Best fit—Place the process into the hole which leaves the small-

est hole left afterwards. In the example, this would be at location 750k.

3. Worst fit—Place the process into the hole which leaves the larg-

est hole left afterwards. In the example, this would be at location 300k.

The simulation program will help you to decide which of these poli- cies is the best.

In the simulation program, the timing of the compaction pro-

cess is under your control and can be triggered when one of the- se two events arise.

Although compaction appears on the surface to be the superior method, it has to be remembered that the processor has to halt

other tasks to achieve this. If the processes are long and compac- tion frequent, a lot of processor time can be spent compacting. (There are various ways of making processes re-locatable, but this topic is outside the scope of this manual).

The fixed partition system was used on the IBM 360s. The vari-

able partition system has been used successfully in many com-

puter systems. However, in modern day operating systems, these methods are no longer used. The techniques of paging, segmen- tation and virtual memory have proved to be superior.