

(21) Application No 8822860.6

(22) Date of filing 29.09.1988

(30) Priority data

(31) 262687

(32) 29.09.1987

(33) IE

(51) INT CL⁴
G06F 12/08 11/34(52) UK CL (Edition J)
G4A AFMD AFN(56) Documents cited
GB 1508731 A GB 1452865 A EP 0239715 A2
EP 0194533 A2 US 4183083 A(58) Field of search
UK CL (Edition J) G4A AFMD AFN
INT CL⁴ G06F

(71) Applicants

Alan Lush
Bellavista, Station Road, Sutton, County Dublin,
IrelandMichael Kenny
176 Fernhill, Arklow, County Wicklow, Ireland

(72) Inventors

Alan Lush
Michael Kenny

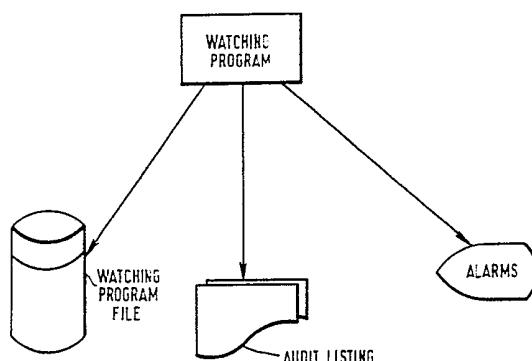
(74) Agent and/or Address for Service

Haseltine Lake & Co
Hazlitt House, 28 Southampton Buildings,
Chancery Lane, London, WC2A 1AT, United Kingdom

(54) Performance-related resource allocation

(57) A computer operating under a virtual memory operating system includes program means for enhancing the performance of the operating system. The program means includes means for gathering job-related information from the system during a user-controllable period of time while the system is running. There is also provided means for analysing the information so gathered to produce recommended adjustments to the operating system parameters, and means for automatically making the recommended adjustments to the operating system parameters, e.g. memory space available to different processes.

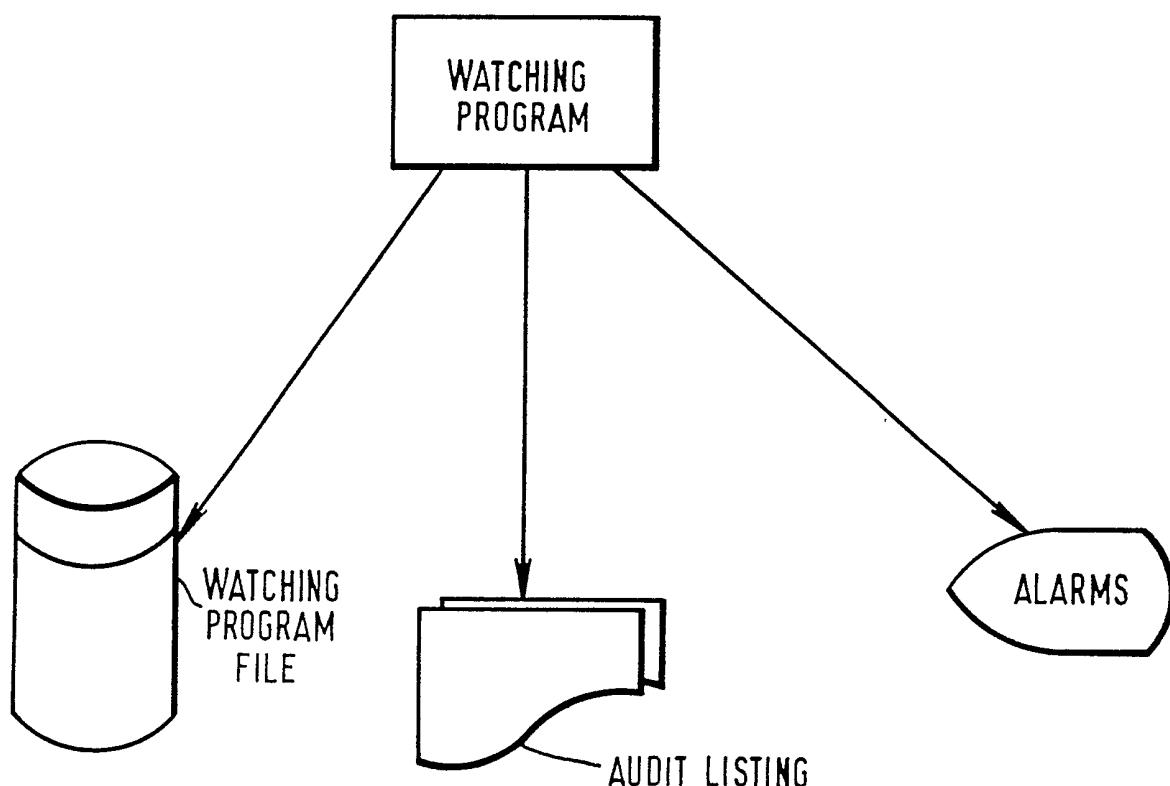
FIG. 1
WATCHING PHASE



USER CONTrollable

- 1 LENGTH OF WATCHING PHASE:
START RUN TIME (E.G. 09.00)
FINISH TIME (E.G. 18.00)
- 2 DURATION OF SAMPLING PERIODS:
STANDARD PERIOD (E.G. 300 SECS)
EVENT RELATED PERIOD (E.G. 10 SECS)
- 3 AUDIT FACILITY ON/OFF
- 4 ALARM FACILITY ON/OFF

FIG. 1
WATCHING PHASE

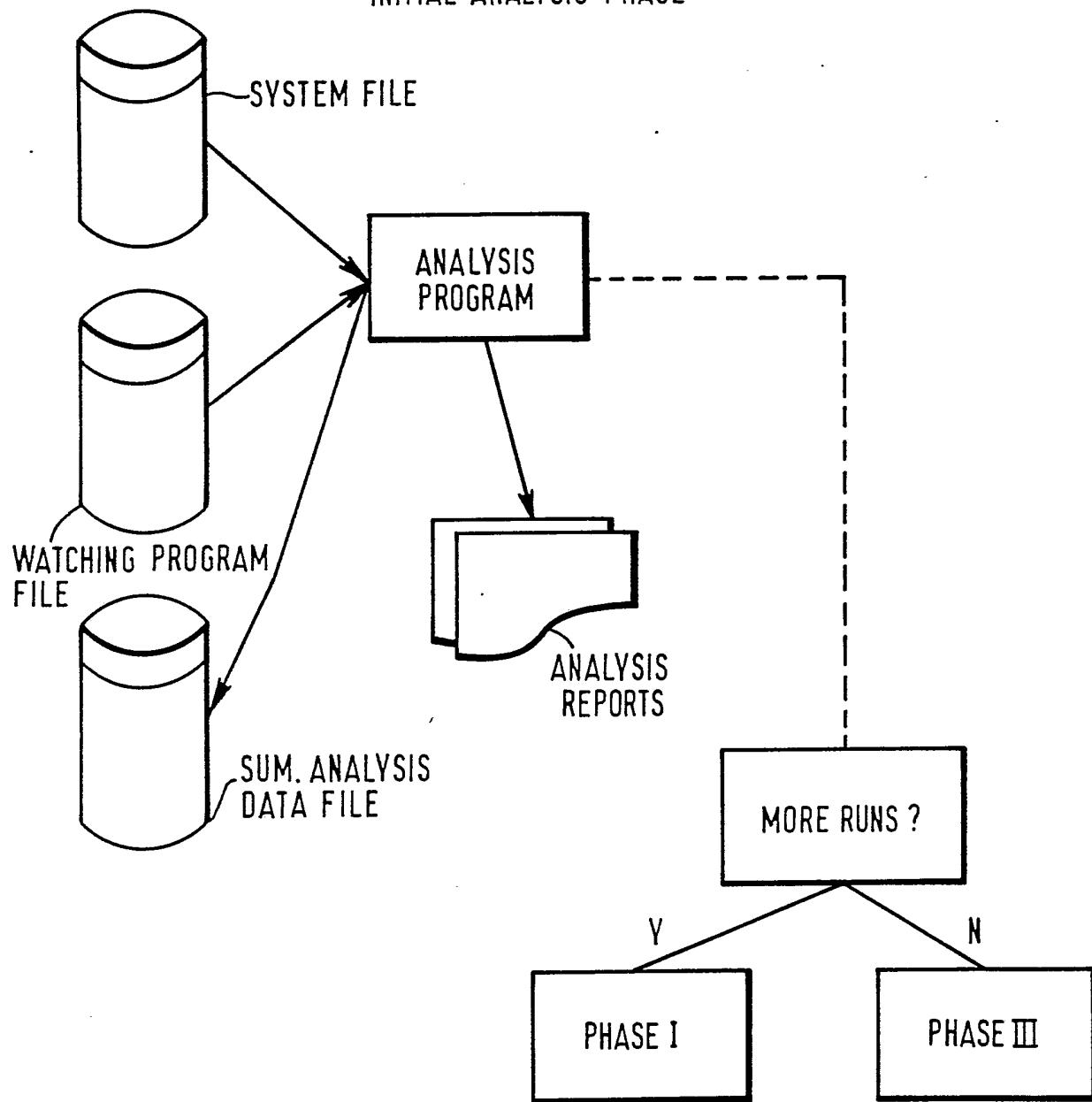


USER CONTROLLABLE.

1. LENGTH OF WATCHING PHASE:
START RUN TIME (E.G. 09.00)
FINISH TIME (E.G. 18.00)
2. DURATION OF SAMPLING PERIODS:
STANDARD PERIOD (E.G. 300 SECS)
EVENT RELATED PERIOD (E.G. 10 SECS)
3. AUDIT FACILITY ON/OFF
4. ALARM FACILITY ON/OFF

2/3

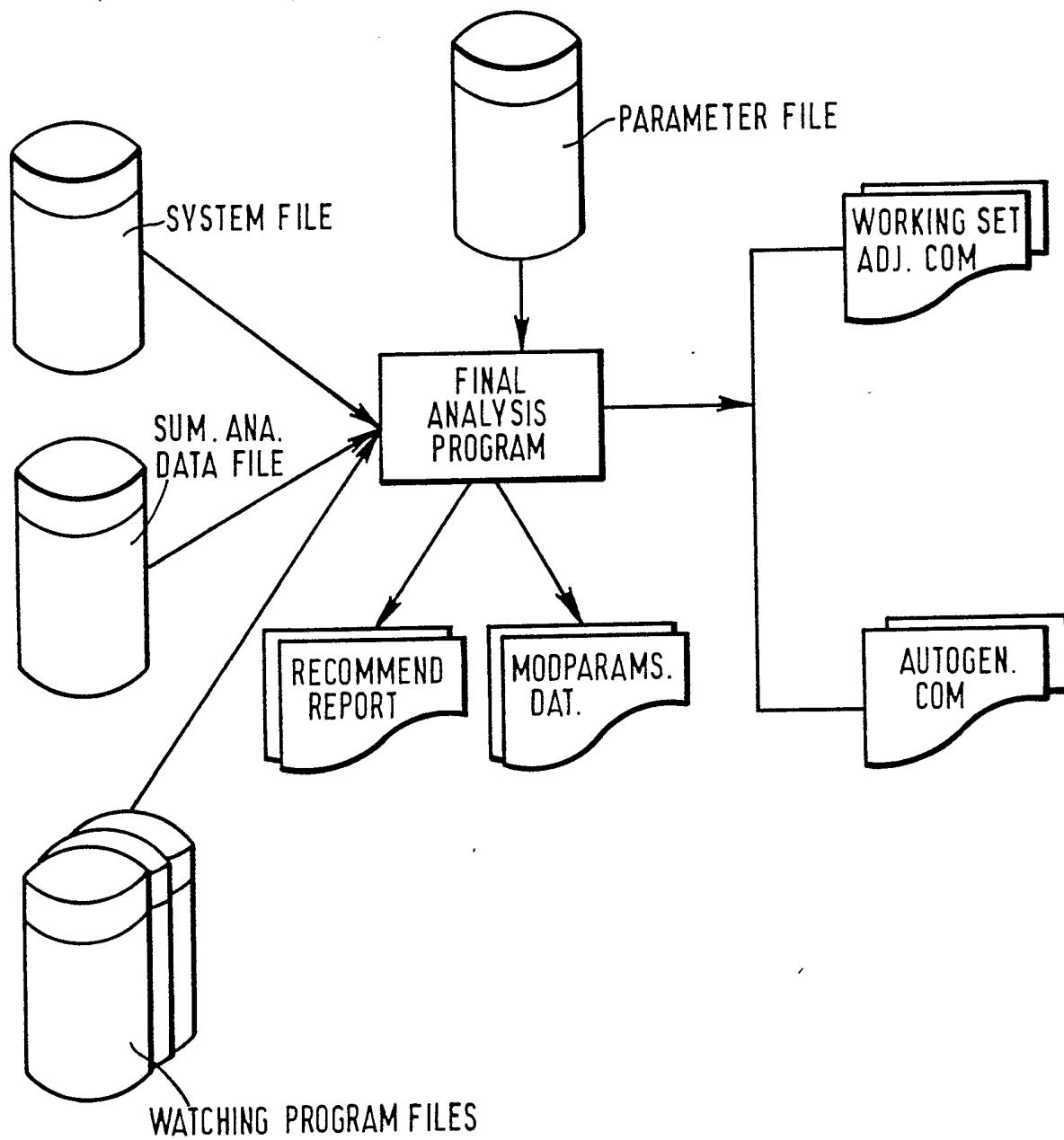
FIG. 2
INITIAL ANALYSIS PHASE



USER CONTROLLABLE

1. THE START TIME FOR THE ANALYSIS
2. THE NUMBER OF RUNS IN THE CYCLE

FIG. 3
FINAL ANALYSIS PHASE



2210462

VIRTUAL MEMORY COMPUTER SYSTEM

This invention relates to a computer operating under a virtual memory operating system, and including program means for improving the performance of the operating system.

5 The invention is especially, but not exclusively, applicable to the VMS operating system which runs on the Digital VAX computer range, and the invention will be described with respect to the VMS operating system using the standard terminology associated with such operating
10 system and which is familiar to those skilled in the art. A description of the VMS operating system parameters which are used in this Specification may be found in, the publication, VAX, VMS system generation utility reference manual AA-Z433A-TE published by Digital Equipment
15 Corporation.

20 The VMS operating system is responsible for the sharing and use of the computer system resources. The main resource is the memory, physical and virtual, of the machine. There are various mechanisms within the operating system to allow a wide variety of user and application types. These mechanisms and their related control parameters will now be briefly described.

VMS Operating System

The operating system provides a multiuser, multiprogramming environment. As with any multiprogramming operating system the resources of the computer must be shared between the various users. The control and management of this sharing is the function of VMS. Among the resources to share is the physical memory of the computer itself. The technique used by the VMS operating system is known as memory management. Virtual memory is a set of locations in physical memory and in secondary storage. The operating system employs a virtual addressing technique to reference the total virtual memory available. The amount of virtual memory available on a machine depends on the size of physical memory and the amount of disk storage used. One of the features of VMS is that only a portion of the image being executed has to be in physical memory. VMS uses a set of procedures, the pager, to ensure that required pages are brought into physical memory when needed.

As stated above, VMS is a multiuser system and therefore it requires some way to identify and control the various users. The mechanism used by VMS is to establish a "process" for the user when access is gained to the system. When setting up a process, VMS will reserve various registers, tables and stacks which will be used by the operating system to identify and schedule the process. The operating system requires some space in physical memory. A portion of this space is set aside for the various processes as they become active.

Other portions of the VMS space contain the executive code, data and routines.

Memory Management

When a program is run a copy of the program is taken from the disk where the program was stored, when it was created, and placed in virtual memory, in a file called PAGEFILE.SYS . This file, although it is also on the disk, is a specially organized file and the program can be thought of as being broken into units of 512 bytes each. A number of these units or "pages" are brought into main (i.e. physical) memory at this time. The instructions on these pages can now be executed but eventually an instruction will be required that is not on one of these pages. To get that instruction, the page containing it must be brought into main memory from virtual memory out on the disk.

15 The condition that arises when a page is required that is not in the main memory is known as a page fault.

When a page fault occurs and there is no room in main memory for the new page, an old page must be taken out so that the new one can take its place. If the page to be removed has not been altered since it was brought in, then we can just overwrite it with the new page. However, if the page has been altered it must be written back out to virtual memory before its location in main memory can be overwritten.

20 25 To reduce the number of I/O's to disk these discarded pages are kept in a special list and later on a cluster of them can be written to the disk in one I/O operation. This special list is called the modified page list. When the number of pages on this list is greater than the sysgen parameter MPW_HILIMIT, pages are written from the list to disk until the value of another sysgen parameter

MPW_LOLIMIT is reached.

Since pages in the modified-page list are still in main-memory it would be faster to get a page from the list than to go out to the disk for it. Therefore when a 5 page fault occurs we can look in the modified page list for the required page before checking the disk. There is a high probability that the page required will be here and since access to a page in the list would be faster than to a page on the disk this is a desirable situation.

10 The free-page list is another list, to which discarded pages that have not been modified are written. Now when a new page is required both the modified page list and the free-page list are searched before the file PAGEFILE.SYS.

15 The parameters which govern the size of the free-page list are FREELIM and FREFGOAL.

If the required page is found in one of the lists the page fault which occurred is known as a soft page fault, whereas if the operating system had to go to the disk to find the page it is known as a hard page fault. ,

20 Working Sets

The set of pages that a process has in main memory is known as the process's working set. The working set size is the number of pages in the working set.

25 A process's working set size is governed by three parameters which are set up for each user. They are WSDEFAULT , WSQUOTA and WSEXTENT.

When a user logs in they are given a working set, the size of which is determined by WSDEFAULT. If the user should need more pages then their working size is increased to WSQUOTA provided there is enough memory 5 available. Should they require still more pages the operating system will increase their working set size to WSEXTENT if there is enough memory available and the amount available is greater than the value in the sysgen parameter BORROWLIM. Also the user's working set size 10 must not exceed another sysgen parameter WSMAX.

The Swapper Trimmer

When the operating system requires additional free pages the swapper trimmer takes a number of steps to get them. Firstly the swapper trimmer looks at the state queues 15 that contain resident processes starting with suspended processes, then those in an LEF state (Local Event Flag - waiting for an external event), and so on.

Any job that it finds in such a state (trimmable job) with a working set size greater than that job's working set quota (WSQUOTA) is trimmed back to its quota. If further pages are required each trimmable job has its working set trimmed back to the value of the sysgen parameter SWPOUTPGCNT. Should even more pages be needed the swapper will swap out entire working sets to 20 SWAPFILE.SYS, which is another file which along with PAGEFILE.SYS makes up virtual memory. Note that a process 25 still in its initial quantum cannot be swapped out.

Automatic Working Set Adjustment

The Swapper Trimmer works on an "as needed" basis - it 30 only trims when pages are needed. Automatic working set

adjustment (AWSA), on the other hand, is time based. Every so often, depending on the value of the sysgen parameter AWSTIME, the AWSA mechanism inspects the page-faulting rates of the jobs on the system. If any jobs 5 are faulting too high their working set sizes are increased in order to reduce the rate of faulting for that process.

A job is faulting high if its fault rate is greater than the sysgen parameter PFRATH (page fault rate high). If 10 this is the case the jobs working set size is increased by the value of another sysgen parameter WSINC (working set increment).

Likewise the working set size of a job whose fault rate is less than PFRATL (page fault rate low) will be 15 decreased by WSDEC (working set decrement).

The following therefore summarises the above operating system mechanisms and their control parameters:

Operating System Mechanisms

AWSA:

20 Automatic working set adjustment.
Based on the page faulting rate of processes, the operating system can increase or decrease the working set size of that user. The working of this area is controlled by the following sysgen parameters:

25 PFRATH Page fault rate high, jobs faulting above this figure will have their working set size increased.

PFRATL The lower rate, jobs faulting below this
rate will have their working set size
decreased.

5 WSINC The amount by which to increase the working
sets.

WSDEC The amount by which to decrease the working
sets.

SWAPPER TRIMMER:

10 Before removing a job from the working set, see below,
the operating system will check each process for pages
which can be removed from the working sets. This
mechanism also checks the modified page list for
possible removable pages. The aim is to free the
required space without major upheaval.

15 **Related Parameters:**

SWPOUTPGCNT The lower limit to which the working set of
a process can be reduced before swapping
will take place.

20 LONGWAIT Used to determine if a process is available
for trimming.

WSQUOTA Initial trimming level.
This figure can be unique for each user on
the machine.

SWAPPER:

25 When the operating system requires additional space or a
high priority job, which is outswapped, becomes

computable the operating system will attempt to remove the working set of a process from the balance set (the total of all working sets in physical memory including the operating system working set).

5 Related parameters:

BALSETCNT The number of different working sets that can be resident in the balance set.

10 DORMANTWAIT This parameter specifies the amount of time a process must be in a computable state before becoming a candidate for swapping.

15 When these mechanisms (ASWA, SWAPPER, SWAPPER TRIMMER) are operational they are using system resources. As resources are limited excessive use by the operating system mechanisms means that the users of the machine suffer in terms of the number of jobs that can be carried out and the speed which they complete.

20 The operating system can be tailored (tuned), by changing the operating system parameters, using either the sysgen utility or a command procedure called autogen. Most, installations adjust the parameters once, when the machine is purchased, using the command procedure provided. The command procedure uses the same calculations for all machines and does not take into account what type or the number of applications that the machine will have to handle. To adjust the parameters effectively requires one to collect information about the operation of the machine, analyse the data and based on experience and knowledge adjust the necessary parameters. 25 There are utilities available with the operating system, which allow one to view activity on the machine, but these do not give any indication of problems (i.e. areas

of inefficiency) or any solution to a problem. They provide one with data which when collated can be used as part of a performance enhancement exercise, but this requires expertise on the part of the user.

5 Accordingly, the present invention provides a computer operating under a virtual memory operating system and including program means for enhancing the performance of the operating system, the program means including means for gathering job-related information from the system
10 during a user-controllable period of time while the system is running, means for analysing the information so gathered to produce recommended adjustments to the operating system parameters, and means for automatically making the recommended adjustments to the operating
15 system parameters.

An advantage of the invention is that it obviates the need for a company to have a VMS expert within the organization. Not only can it identify problem areas within the operating system it also provides the necessary adjustment to the relevant parameter to alleviate the problem. Having calculated the new value the invention can make the change to the parameter thereby allowing the user with very little VMS knowledge to get the most from the machine.

25 For the user with some knowledge who wants to learn more, the invention can be designed to provide a verbal explanation of the steps taken to arrive at any new values. These explanations are produced in the form of a hard-copy report. If the user wants to analyse the activity on the machine a comprehensive report section
30 may be provided. Apart from the tuning aspect of the invention, other useful interactive resource management

features may be provided.

	Alarm system	Used to highlight processes hogging the system.
5	System usage display	Shows the users currently active on the machine. Information displayed includes username, process name, state of the process and the image currently active.
10	Load Analysis	Using bar-chart format the page file and swap file usage is displayed. This analysis also shows the cpu usage and the amount of unused space on the system disk.
15	The invention may be menu driven, for ease of use, and have its own built-in help facility.	In an embodiment of the invention, the program consists of three modules corresponding to respective phases, referred to below, and runs detached and independent of the user. There is a facility for communication between the user and the detached process under menu control. To overcome any problem caused by using information gathered at peak work load times the program gathers data over a number of runs, e.g. one run on each of a number of days. The number of runs, known as a cycle, and the duration of these runs (e.g. from 09.00 to 18.00 hrs in each day) is controlled by the user. The facility also exists to schedule the analysis phase so as to avoid times when the machine is heavily loaded.
20		
25		

In a typical cycle, there are the following elements:-

5 PHASE I (Watching Phase). During this phase the watching module observes the system at regular intervals and records information about all processes currently active
10 in data files. The watching module has the ability to adjust the sampling period used, depending on the events it encounters. During uneventful periods using a Standard Sampling Period of e.g. 300 seconds the program is therefore less active than at busy times when it is
15 very active and uses an Event Related Sampling Period of e.g. 10 seconds. An audit facility is provided so that every record created by the watching program can be relayed to the user, and so that certain specific events can be advised to the user by way of an alarm to the
20 user's terminal. This enables the user to respond at once to that problem. At start time, the user has the facility to provide values for six control parameters (Figure 1) that govern the watching program. During running, these can be altered as needed by use of a communications menu.

25 PHASE II (Initial Analysis Phase). When the monitoring phase is complete, control is passed to the analysis module for analysis of the recorded file. The findings of the analysis are held, and are used later in the major analysis at the end of all programmed runs. This mini analysis would typically take place at the end of a day of monitoring and provides the user with useful information about the day's activity both by username and image name, as well as providing information to the major
30 analysis module described below.

When this phase is complete, the program can either repeat the watching phase, or move to phase III, final

analysis. This decision is based on the number of runs called for in the overall cycle which was input by the user at start time. The start time for the analysis is also under user control.

5 PHASE III (Final Analysis Phase). This module uses the findings of the individual analysis to identify problem areas on the machine. The module will calculate new values for sysgen parameters taking into account the environment of the machine. In addition to looking at 10 the sysgen parameters the module also examines the working set characteristics of the users that were page faulting above average. Based on an analysis of the working set usage the module will calculate better values for that user. If any changes are to be made to working 15 set values for users a command procedure will be produced which contains the adjustments and the necessary commands to invoke the system utility Authorize. Any changes to system parameters are written to a system- provided file. The user has a choice whether to allow the program to 20 make the changes automatically, in which case the program will invoke the command procedure autogen which will use the file containing the changes.

Briefly, in the case of the swapper trimmer, the program operates as follows:

25 During the individual runs the program gathers information related to swapper trimmer activity. It examines the finds looking for any process where page faulting was incurred following activity by the swapper trimmer. The program analyses these occurrences to 30 evaluate whether swapper trimmer activity is excessive

and a cause of greater operating system activity. If this is found to be the case, the program will adjust the related parameters to reduce the activity of the trimmer.

5 As previously discussed, the trimmer will first reduce the working set size of a process to the value of the parameter WSQUOTA, and then if more space is required the working set can be further reduced to the value of the parameter SWPOUTPGCNT. Another parameter is available, LONGWAIT, which is used by the trimmer to determine if a
10 process has been in a particular state for an acceptable length of time before trimming is attempted.

15 To reduce the amount of swapper trimmer activity, the program will attempt to increase the value of the parameter, LONGWAIT, up to an in-built limit, a recognised upper operational limit. If that parameter cannot be increased the program will increase the value of the SWPOUTPGCNT parameter. Increasing this parameter would have the effect of decreasing swapper trimmer activity while increasing swapper activity. The first
20 level of increase is to change the parameter to equal the typical working set size on the machine. During the, observation of the system the program calculates the typical values for each installation. If it finds that the parameter is already greater than the typical working
25 set size the program will attempt to set the parameter equal to a typical working set quota. If this is not possible, or has already been done the program will adjust the value of SWPOUTPGCNT to completely switch off the trimmer mechanism.

30 An embodiment of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 illustrates the watching phase of the invention,

Figure 2 illustrates the initial analysis phase, and

Figure 3 illustrates the final analysis phase.

This embodiment of the invention comprises a computer
5 program running on any Digital VAX computer operating
under VMS. The program will be described in the form of
flow charts in sufficient detail to enable a programmer
skilled in the art of VMS to implement the invention.
Before giving the flow charts, however, the contents of
10 the various files used in the three phases shown in the
drawings are as follows:

<u>FILE NAME</u>	<u>CONTENTS</u>
1. System File	(a) Security Related information (b) Run Parameters (c) Machine dependant values
5 2. Watching Program File	There are 2 types of records produced by the watching program
	Type 1. Sample Record
10	(a) Sample Number (b) Total Amount of Direct and Buffeted I/O (c) Number of jobs found in various states e.g. COM, COMO, SWAPPED (d) System related totals e.g. Free Memory (e) Memory Usage Flags
	Type 2. Detail Record
15	(a) Sample Number (b) Username/Imagename (c) Working Set characteristics/usage (d) State of the process (e) Page faulting activity
20	This file is used by initial analysis and main analysis program.
25	3. Parameter File (a) Parameter Name (b) List of related Areas (c) Current Value (d) Original Value

(e) Recommended values are 5 cycles

- 4. Recommendation
 - File
 - (a) Record Type
 - (b) Original Values for Parameters (max 6)
 - (c) Summary totals from analysis (6)
 - (d) New Parameter Values
- 5
- 5. Summary Analysis
 - File
 - (a) Record Type
 - (b) Run Number
 - (c) Summary totals from initial analysis
 - (d) U(6) sename

WATCHING PHASE (FIG.1)

1. Create mailbox used for communications with menu program.
2. Open files.
3. Check validity of program.
- 5 4. Initialize variables
5. Check Start Run Time established by user.
Wait if necessary
6. Check if audit listing required.
If yes then open audit file.
- 10 7. Establish main control loop for duration of the run.
8. Main Loop (Detailed below).
9. Produce Summary record.
Write Summary record to Watching program file.
10. Check start time for initial analysis.
- 15 If delay then wait.
11. Chain to Initial Analysis Program unless user instructed to stop run now.

MAIN LOOP (Stage 8 in Watching Phase)

1. Sleep for Standard Period (Fig. 1).
2. Clear out counters.
3. For every process on the machine:
 - 5 3.1 Initialize variables.
 - 3.2 Use system services to get process information.
 - 3.3 Check if new process. If new process store PID, clean out accumulators.
 - 3.4 Remove Directory/Device Specification from Image Name.
- 10 3.5 Calculate current values for memory usage.
- 3.6 Store information to Watching program data file.
- 3.7 Accumulate Sample Totals, e.g. number of jobs, number of jobs in various states.
- 3.8 Write line to Audit Trail listing if Audit flag is on.
- 15 3.9 Move current values to running totals.
4. Write sample Summary record.
5. Send message to user for system "Hogger" if the alarm system is on.
6. Send message to user for swapping activity if alarm system is on.

7. Write line to Audit Listing file if audit facility is on.
8. Set flag if any events high-lighted within samples, e.g. swapping, high page faulting, etc.
- 5 9. Check event flag setting.

If flag is set then repeat steps 3 to 8 using Event Related Period (Fig. 1).
10. Check for any messges from user.
11. If any message sent, take appropriate action - send back acknowledgement.

INITIAL ANALYSIS PHASE (FIG. 2)

1. Open Files
 - (a) System file
 - (b) Watching Program Data file
 - 5 (c) Summary Analysis file
 - (d) Hardcopy Report file
2. Initialize variables.
3. Do Analysis (detailed below).
4. Calculate typical working set values.
- 10 5. Check if end of cycle (any more runs?). If yes then go to Final Analysis.
6. Calculate length of wait before start of watching phase.
7. Wait.
- 15 8. Go to watching phase.

ANALYSIS (Stage 3 in Initial Analysis Phase)

1. Extract control record from Watching Program data file.
Get number of samples taken.
2. Write number of samples to Summary data file.
- 5 3. Calculate maximum number of jobs found in the Balance Set and the maximum number of processes found during the Watching Program.
4. Analyse Image Activation (Detailed below).
5. Analyse Page Faulting Activity.
- 10 6. Analyse Page Faulting Activity for sub-processes.
7. Analyse Swapper Trimmer activity (Detailed below).
8. Analyse Swapper Activity.
9. Analyse AWSA mechanism.

ANALYSIS IMAGE ACTIVATION (Stage 4 in Analysis)

1. Set up loop for each sample taken.

1. Set up inner loop for each process found in each sample.

5 1. Get record

2. Using index check Username.

3. If Username is different then write record from cumulative information.

Set up new variables.

10 4. Store data in analysis variables i.e. image count and CPU usage.

ANALYSIS SWAPPER TRIMMER ACTIVITY (Stage 7 in Analysis)

1. Set up loop for each sample.
 1. Get control record from Watching Program Data file.
 2. Check if any Trimmed jobs. If not then go to end of loop.
 3. Set up inner loop for each process in the sample.
 1. Get detail record from Watching Program Data File.
 2. Count if faulting above Defined Page Fault Rate and the Job is trimmed.
- 10 4. Cumulate totals for sample:
 - (a) Number of jobs
 - (b) Number of trimmed jobs
 - (c) Number of High-Fault jobs
 - (d) Number of trimmed jobs that are High-Faulting
- 15 2. Write Summary Record to Final Analysis file containing cumulated totals.

NOTE: It will be appreciated that the other mechanisms referred to in stages 5, 6, 8 and 9 will be similarly analysed as the examples given above.

FINAL ANALYSIS PHASE (FIG. 3)

1. Open Files

- (a) System file
- (b) Summary Analysis file
- 5 (c) Parameter Value file
- (d) Recommendations file

2. Initialize variables

3. Calculate typical Working Set values over each run in the cycle.

10 4. Total the number of samples in the cycle.

5. Analyse Page Faulting.

6. Analyse Page Faulting for sub-processes.

7. Analyse Swapper Trimmer performance (detailed below).

8. Analyse Swapper activity.

15 9. Analyse ASWA mechanism performance.

10. Analyse occurrences of low memory availability.
11. Analyse existing settings for major system parameters.
12. Check if automatic changing is requested.

5 If yes then modify the system file "MODPARAMS.DAT" to reflect desired parameter changes.
13. Check if any changes to Working Set values.

If yes then start sub-process using command procedure produced during analysis.
14. Check if any changes to System Parameters.
- 25 If yes then start sub-process using command procedure produced during Analysis.
15. End.

ANALYSIS SWAPPER TRIMMER PERFORMANCE (Stage 7 of Final Analysis)

1. Initialize variables.
2. Cumulate totals from each relevant summary record in data file.
- 5 3. If number of jobs trimmed is less than 20% of total number of jobs and the number of trimmed and faulting jobs is less than 20% of the total of faulting jobs then there is no recommendation.
(End of Analysis section)
- 10 4. If the current value of the system parameter LONGWAIT is less than 20 then increase this parameter in steps of 5 with a maximum value of 20.
(End of Analysis section)
- 15 5. If the current value of the system parameter SWPOUTPGCNT is less than the typical Working Set size then increase the value of the system parameter.
(End of Analysis section)
- 20 6. If the value of the system parameter SWPOUTGCNT is less than the typical Working Set Quota then increase the value of the system parameter.
(End of Analysis section)
7. Set the value of the system parameter SWPOUTGCNT to zero. This has the effect of switching off the Swapper Trimmer mechanism.
- 25 The increases of changes to system parameters are recorded in the Recommendation file. If automatic changing has been selected then the program will write the new values to the file utilised by the system, "MODPARAMS.DAT".

CLAIMS

1. A computer operating under a virtual memory operating system and including program means for enhancing the performance of the operating system, the program means including means for gathering job-related information from the system during a user-controllable period of time while the system is running, means for analysing the information so gathered to produce recommended adjustments to the operating system parameters, and means for automatically making the recommended adjustments to the operating system parameters.
5
2. A computer as claimed in Claim 1, wherein in order to free space in physical storage for pages to be transferred thereto from external storage the operating system includes a first mechanism (SWAPPER TRIMMER) for reducing the working set of at least one process which has been inactive for a period at least equal to a first operating system parameter (LONGWAIT) progressively down to sizes determined by second and third operating system parameters respectively (WSQUOTA and SWPOUTPGCNT), and a second mechanism (SWAPPER) for transferring the entire working set to the external storage if the first mechanism does not free enough space, wherein the analysing means is responsive to the gathered information to determine, according to a pre-determined criterion, whether a decrease in the level of activity of the first mechanism (SWAPPER TRIMMER) is desirable and, if so, to increase the first operating system parameter (LONGWAIT) up to a pre-determined system maximum, or if the first parameter is already at the maximum, to increase the third operating system parameter (SWPOUTPGCNT) up to a value related to the individual second operating system
10
15
20
25
30

parameters (WSQUOTA) of all working sets, or if the third operating system parameter (SWPOUTPGCNT) is already at or above the lastmentioned value, to reduce the third operating system parameter (SWPOUTPGCNT) to zero, and wherein the automatic adjustment means records the new values of the first (LONGWAIT) or third (SWPOUTPGCNT) system parameters into a file accessible by the operating system.

3. A computer as claimed in Claim 1 or 2, wherein the 10 information-gathering means is adapted to sample the activity of the operating system at intervals which are dependent upon the intensity of such activity.

4. A computer as claimed in Claim 1, 2 or 3, wherein the 15 information gathering means is adapted to gather information during a number of separate runs each of a user-controllable duration, and the analysing means is adapted to perform an initial analysis after each run and a final analysis after all runs.

5. A program-controlled method for enhancing the 20 performance of the operating system in a computer operating under a virtual memory operating system, including gathering job-related information from the system during a user-controllable period of time while the system is running, analysing the information so gathered to produce recommended adjustments to the operating system parameters, and automatically making the recommended adjustments to the operating system 25 parameters.

6. A method as claimed in Claim 5, wherein in order 30 to free space in physical storage for pages to be transferred thereto from external storage the operating

system includes a first mechanism (SWAPPER TRIMMER) for reducing the working set of at least one process which has been inactive for a period at least equal to a first operating system parameter (LONGWAIT) progressively down to sizes determined by second and third operating system parameters respectively (WSQUOTA and SWPOUTPGCNT), and a second mechanism (SWAPPER) for transferring the entire working set to the external storage if the first mechanism does not free enough space, and wherein the method comprises analysing the gathered information to determine, according to a pre-determined criterion, whether a decrease in the level of activity of the first mechanism (SWAPPER TRIMMER) is desirable and, if so, increasing the first operating system parameter (LONGWAIT) up to a pre-determined system maximum, or if the first parameter is already at the maximum, increasing the third operating system parameter (SWPOUTPGCNT) up to a value related to the individual second operating system parameters (WSQUOTA) of all working sets, or if the third operating system parameter (SWPOUTPGCNT) is already at or above the lastmentioned value, reducing the third operating system parameter (SWPOUTPGCNT) to zero, recording the new values of the first (LONGWAIT) or third (SWPOUTPGCNT) system parameters into a file accessible by the operating system.

7. A virtual memory system substantially as described with reference to the accompanying drawings.