



OPERATING SYSTEMS





Synopsis

I - Introduction to the selected Operating System	- 3 -
History.....	- 3 -
New features.....	- 4 -
Equipment required.....	- 4 -
Several interfaces	- 5 -
Unwanted.....	- 5 -
Such a waste.....	- 5 -
 II – Process Control Management	 - 6 -
 III – Deadlock Management	 - 8 -
Definition	- 8 -
Conditions.....	- 8 -
Strategies	- 9 -
Ostrich Algorithm.....	- 10 -
Detection and recovery	- 10 -
Prevention	- 10 -
Avoidance	- 10 -
Windows Vista.....	- 11 -
 IV – Memory Management.....	 - 13 -
Definition	- 13 -
Compaction	- 14 -
Memory allocation strategies.....	- 14 -



Internal fragmentation.....	- 15 -
Memory partitioning.....	- 15 -
Fixed memory partitioning	- 15 -
Variable memory partitioning	- 16 -
Virtual Memory	- 16 -
Page replacement algorithm.....	- 16 -
Windows Memory Management.....	- 17 -
SuperFetch	- 17 -
ReadyBoot	- 17 -
ReadyBoost	- 18 -

V – Secondary Disk Scheduling Management	- 19 -
FCFS scheduling	- 19 -
SSTF scheduling	- 20 -
SCAN.....	- 21 -
C-SCAN	- 22 -

No	Student Name	Research and Investigation (30)	Installation(20)	Documentation (10)	Referencing (10)	Analysis (15)	Presentation (15)	Total (100)
1								



I - Introduction to the selected Operating System

History

We are starting our project in Operating systems; we have to choose one operating system. I have chosen Windows Vista. Let me introduce this operating system to you. Vista is an operating system created by Microsoft and was launched on the market 5 years after Windows XP. Windows Vista is a real disappointment to everyone, Microsoft is already working on a new operating system (Windows Seven) with less bugs and others problems.

According to Microsoft, Vista has about a hundred new features like a new graphic interface or a new DVD creator. The first goal of Microsoft was to build a better security on the new operating system because the XP has a really bad protection against virus. At this time, Microsoft is not talking about his creation Vista but Microsoft is betting on his successor "Seven" which is supposed to make Vista's defaults forget.

The final version of Vista was officially presented the 30th of January 2007. This took a very long time because Microsoft has to correct the stability, the security problem and many bugs. Vista had announced a new revolutionary management of files called "Win FS" but it was not ready and too slow so they let it down.



New features

Here are some new features available in Vista:

- Interface Aero, which make the interface pleasant and also present many advantages.
- UAC (User Account Control), we can choose to make an account administer with all the access or to make an account for visitors with few possibilities.
- Direct X 10, it's a new version a Direct X 9 which is only available in Vista. Direct X is used for video games.
- Internet Explorer 7, which can show several pages in the same window.
- Windows Update, it will look up for you the newest pilotes.

Equipment required

Windows Vista requires a recent and a powerful equipment to be installed. Microsoft has identified 2 sorts of computers for Vista:

- Vista Capable
- Vista Premium Ready

The computers in the Vista Capable's category need to have a processor at least 800 MHz, 512 Mo of RAM and a graphic card able to carry Direct X 9. These computers won't be able to show all the graphic features of Vista.

The computers in the Vista Premium Ready's category need to have a processor at least 1GHz, 1Go of RAM and a graphic card able to carry Direct X 9. These computers will be able to show all the graphic features of Vista and also use the Aero interface.



Several interfaces

Vista is not famous for his performance but for his design. Vista is really pleasant, Microsoft decided to let the client choose between 3 interfaces:

- Aero, which contains 3D, animations and special effects. Aero is used by performing computers.
- Basic, which doesn't have 3D but can play animations and special effects of Windows XP. Basic if for computers which don't have enough power to use Aero.
- Classic, which doesn't have special effects and animations. His interface looks like Windows 2000. It doesn't use much memory.

Unwanted

- Some tests on Vista showed that the autonomy of the computer was reduced between 10 and 30% with Aero.
- The equipment required is too high for many people. So if you want to install Vista, you have to buy a new computer and that's expensive.
- The operations with files are much longer, when you want to copy and delete something you often have to wait a bit.

Such a waste...

Many users of Windows were waiting for this new operating system and put a lot of hope in Vista. New features, better security, a new interface, a waiting period of 5 years, many companies decided to change XP for Vista, Microsoft deceived so much people with this operating system.



II – Process Control Management

After having done the introduction of Windows Vista, we have to know what kind of CPU we are dealing with. Scheduling is a concept really important in computer multitasking and in multiprocessing design. Scheduling is the way processes are designed to run on the CPUs. We have seen in class that we have 2 kinds of algorithms:

- Pre-emptive
- Non pre-emptive

For the pre-emptive one, we have the RR (Round Robin) and for the non pre-emptive we have 3 choices: FCFS (First-Come First-Served), SJF (Shortest Job First) and Priority. The goal is to know if Windows Vista uses a pre-emptive or a non pre-emptive scheduling. Let's explain what exactly is a pre-emptive multitasking and a non pre-emptive multitasking.

Pre-emptive multitasking is a task where the operating system decides how long a task needs thanks to information before giving the hand to another task. Pre-empting is an action where you get the operating system under control and you decide to change the task. This algorithm can decide to stop a process whenever it wants. The time that a task needs is called the "elapsed time". The process which is controlling the CPU has to offer control to the other processes. All programs need to work together. That's why pre-emptive multitasking is also called "Cooperative multitasking".

Non pre-emptive multitasking is a style of multitasking where the operating system never chooses to go from a process to another. It's the process itself which decides to interrupt and gives control to another process. For this case, the process can't be interrupted until it's done. The process will be done if it is executed successfully.

Let's get back in history; Windows first used MS-DOS which was a non-pre-emptive algorithm. The system was not even multitasking. Then, we have Windows 95 which was a pre-emptive scheduler. After, we have Windows NT which used a multilevel feedback queue. Then, we had Windows XP and Vista which also used this multilevel feedback queue.



MS-DOS	Windows 95	Windows NT	Windows XP	Windows Vista
Non pre-emptive	Pre-emptive	Multilevel Feedback queue	Multilevel Feedback queue	Multilevel Feedback queue

So let's get interested in the Vista case. Vista uses a multilevel feedback queue. Multilevel feedback queue is a scheduling algorithm. He prefers to treat short jobs first, that means that he will treat tasks which don't need much resources. And also Multilevel feedback queue gives priority to input and output devices and treat them first.

The multilevel feedback queue is a combination of round-robin, First in First Out (same as FCFS, First-come First-served) and finally Priority. The Round robin scheduling is used for high priority processes and the FIFO is used for low priority processes.

Multilevel feedback queue algorithm can make a process move between queues. It can assign different priority to each queue. This algorithm takes the highest priority queue and run that process. This algorithm is used to avoid "starvation". How does it work? If the process is using too much CPU time, it will be removed to lower priority queue. Same, if a process is waiting too much time in the lower priority queue, it will be removed to a highest priority queue. We can count 32 levels of priority, 0 being the lowest priority and 32 the highest priority. We can say that between 0 and 15 it's called a normal priority, after 15 it's a real time priority and these are the ones which are going to be treated first.

How can we determine a priority of a process? First, we call a new process created a "thread". The thread is born with a priority level and will also have a level of priority being a process. So, we can count 2 levels of priority to determine a priority of a process: the thread's priority and the process's priority.

To conclude, Windows Vista is using a combination of pre-emptive (Round Robin) and non pre-emptive (FCFS and priority) algorithms to make the CPU work, which is called the multilevel feedback queue.



III – Deadlock Management

Definition

We call deadlock in operating system a wait state where different processes cannot get their requested resources. Deadlock is a permanent blocking of processes which compete for the communication of the system. It's an unexpected situation that takes place where the process P1 requires a resource R1 which is currently held by the process P2 while the process P2 requires a resource R2 which is currently held by the process P1. We can see two types of waiting: when you have to wait for access to a critical section and when you have to wait for a non preemptive resource.

Conditions

We can count 4 conditions for a deadlock to occur:

- Hold and wait
- Circular wait
- Mutual Exclusion
- No pre-emption

Hold and wait: resources are required by processes which are carrying what they have. Each process can request more than one resource that's why a process is waiting for more resources; this can bring us to a "deadlock".

Circular wait: It's a circular waiting; it's like a chain where each process is waiting for a resource being carried by the next process in the circular chain. This is called the wait state; this situation may also lead to "deadlock".

Mutual exclusion: resources can't be shared. That means that only one process is able to use one resource at a time. The others processes have to wait their turn until they get the resource. This resource has to be in "exclusive mode" to be used one at a time and not in ordinary mode.

No pre-emption: the current process can't take non pre-emptable resources.



Strategies

There are strategies to deal with deadlock:

- This not an easy task. When we have to deal with a deadlock situation, the operating system is calling a “technical person” who is going to resolve the deadlock problem.
- This is a complex method. The operating system handles itself the deadlock.
- This approach is preferred. The operating system ignores the deadlock, it's like the operating system has never been in deadlock situation.

When we have to deal with a deadlock situation, the operating system breaks the deadlock with:

- Process termination
- Resource pre-emption

The process termination is a risky recovery method because it can affect the system. When you terminate a process which is using many resources, this can affect badly the system.

The resource pre-emption, the goal is to terminate one resource at a time. The resources are selected by their priority and time consumption; we also have to avoid starvation during the selection.

We can count 4 strategies to deal with deadlock:

- Ignore the problem
- Detection and recovery
- Prevention
- Avoidance



Ostrich Algorithm

We use this algorithm only if deadlocks are rare; this algorithm is a strategy of ignoring the problem. This may be a solution because the problem is rare. This method is used with deadlocks in Windows.

Detection and recovery

We use this method when a resource is requested, so we update the graph of the resource. We terminate the process if there is a circle; we do that until the circle is broken.

Prevention

Let's remember the 4 conditions for a deadlock to occur:

- Hold and wait
- Circular wait
- Mutual Exclusion
- No pre-emption

If one of the four conditions is not valid, the deadlock doesn't occur. So the prevention is the method where we make sure that one of the 4 conditions is not satisfied.

Avoidance

We call this the Banker's algorithm. This method consists in avoiding actions which lead to deadlock. The method sees if the resource can be treated and then can avoid the deadlock by making the right choice.



Windows Vista

Since the beginning of windows software, it seems that each new operating system is getting better and better: Windows 95, 98, Millennium, 2000, XP, Vista, Seven... The version of the operating is more stable with the time. We can say that Windows Vista is in the continuous of Windows operating system because Vista has some new technologies to prevent errors and recover them. These are the power of the good stability of Windows Vista. The good stability of the operating system is due to four software:

- **Driver verifier** – It searches the circles in the program by using resources graph. If it finds a circle, the “!deadlock” is coming and it is the kernel debugger which is going to stop it before it occurs.
- **Dr. Watson debugging tool** – it searches output and input; when it blocks (using by another user) it cancels the demand. So it sends an error message. It is called the windows error reporting dialog box. If you have Windows, you will notice that many errors occur during your work in the computer. You can see this window appear:



This is windows error reporting dialog box. You can complain directly to Microsoft and ask questions but you will never get an answer. This is where the deadlock appears, it is the one which appears to stop a process when there is a bug. So you can see that deadlock is really important in operating system such as Windows Vista.



- **Reliability monitor** – it is an interface where the user can stop the process himself when he sees a problem. It's like a better task manager.
- **Service recovery** – In vista, each task has a recovery. When there is a problem, the service reboots only the task in question and the tasks connected. The problem is solved.

Microsoft has also created a help for Windows Vista users, it is called the windows error reporting service. It proposes you a solution to your problem.



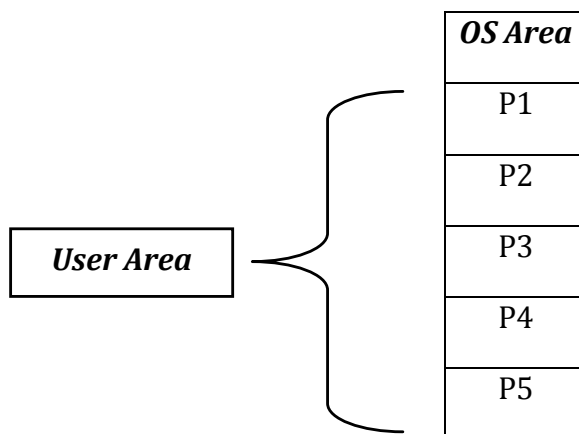
IV – Memory Management

Definition

The memory is decomposed into partitions. We can count 2 main areas:

- **OS Area**, this is the top position of the memory.
- **User Area**, this is the remaining area of the memory.

All the processes are leaded into user area. We can see clearly with a schema:



There are also exchange between the hard drive and the memory:

- **Swap in**, all processes are located in the backing store (HD) and when there is a demand from the main memory, the process demanded will be loaded from the backing store into the main memory.
- **Swap out**, when a process is not wanted in the main memory, it is sent from the main memory into the backing store.



Compaction

It's a technique used to merge all the free holes to one big free hole. This big free hole is available for the other processes. We can see many swap-in or swap-out operations during the compaction operation. So this operation is a time-consuming technique. Compaction can be decomposed in three phases:

- First, we determine the distance that the block is moving
- For each block we have a specific pointer. Then, each pointer must be updated by changing the old location by the new location.
- Finally, the data is moved.

Memory allocation strategies

We are dealing with dynamic memory allocation; the memory has to be saved when blocks of memory are used. We can see three main strategies to allocate space:

- **Best-fit**, this algorithm is searching in the entire memory for the smallest free hole which can contain the process. The problem for this algorithm is that it is a long research: we search in the entire free list to find the free hole which has enough space for the process. The other problem is that this algorithm lets many little holes which are not big enough for process to fit in. We call this situation "fragmentation"; it's a big problem for the best-fit algorithm. We can find a solution for these problems by giving a bigger block than the size requested.
- **First-fit**, this algorithm is searching for a free hole big enough for a process. This algorithm is preferred than the best-fit because there are less fragmentation with the first-fit algorithm. The problem with this algorithm is that there is an accumulation of small holes, so the research will be made further in the memory.
- **Worst-fit**, this algorithm is a theory level, it is never used. It searches the biggest free hole in the entire memory for the process. So that another process will be able to user the big free hole.

First-fit and Best-fit are the best algorithms for memory allocation strategies.



Internal fragmentation

Internal fragmentation is when a memory is allocated and not used. This memory space is wasted. Internal fragmentation is the difference between the free memory space and the size of the process:

$$\text{IF} = \text{Free Space} - \text{Size Process}$$

Example:

$$\text{IF} = 300\text{Kb} - 298\text{Kb} = 2\text{Kb}$$

We have to note that the compaction technique can't eliminate the internal fragmentation.

Memory partitioning

There are two types of memory partitioning:

- Fixed memory partitioning
- Variable/Dynamic memory partitioning

Fixed memory partitioning is for process using a number of pages determined before whereas variable memory partitioning is for process which allocate pages during time of execution.

Fixed memory partitioning

The memory is partitioned into equal sized portions which hold only one process. If there is a free partition, a process can be loaded in that partition. If the partitions are busy, we use the swap-out technique to make some place for the new process. Any program settles in an entire partition, the fixed memory partitioning may lead to "internal fragmentation". Free holes are created and they can't be used anymore.



Variable memory partitioning

When a process is coming, a hole big enough to contain the process is allocated. Free holes are scattered in the memory. Free holes can be combined to make a big block of memory. Variable memory partitioning can lead to “external fragmentation”.

Virtual Memory

A disk contains information; blocks of information are taken from the disk and put into the memory of the processor. We need to know the size of the block of information before removing it. There are 2 ways to determine the size:

- Swapping
- Paging

Virtual memory can let the user run a program whose size is bigger than RAM size. A partition of the hard drive is used as a memory. The virtual memory doesn't exist physically; it's a technique which gives the user the illusion that he is using a program whose size is bigger than RAM size.

Page replacement algorithm

There are 3 main algorithms:

- FIFO (First in first out) – The idea of this algorithm is to put the most recent arrival at the back of the memory queue and the earliest one in the front. This algorithm has the advantage to be cheap.
- LRU (Least recently used)
- Optimal



Windows Memory Management

Memory management in Windows has evolved with the time. It became a complicated architecture. This memory can manage many features and capabilities. Windows Vista has improved in certain domains such as dynamic system space, robustness and section access. Vista lets us use dynamic address space allocation. The Windows Vista gives the user the possibility to have three new features for memory management:

- SuperFetch
- ReadyBoot
- ReadyBoost

SuperFetch

Previously in Windows XP, PreFetching was used to improve startup and boot performance. But Windows Vista uses a better technology: SuperFetch. It's a memory management which remembers chronological information to access this information quickly. SuperFetch can access to external units and use them as a cache memory. This algorithm is used to improve opening speed time and also software the most used. So the system analyses which programs are the most opened each day so that it can open it faster the same day. This algorithm can also use a USB key as an additional RAM memory.

ReadyBoot

If the system has got less than 512 Megabytes of memory it uses the same preFetching boot time than Windows XP. But if it has more megabytes, the boot can be improved. The ReadyBoot has to tell the cache that it doesn't have to change even if data is read or written during the boot. ReadyBoot permits to win about twenty percent in terms of performance comparing to Windows XP preFetcher.



ReadyBoost

Windows Vista offers a new technology of cache disk: ReadyBoost. The goal of the ReadyBoost is to improve the waiting time for flash memory such as USB key, SD card, etc... Vista is about ten times faster with this new technology. Vista is using the free space in the memory flash as a cache data. It's the SuperFetch, whose we talk about sooner, which uses this free space to make the waiting time shorter and make the treatment of applications faster.



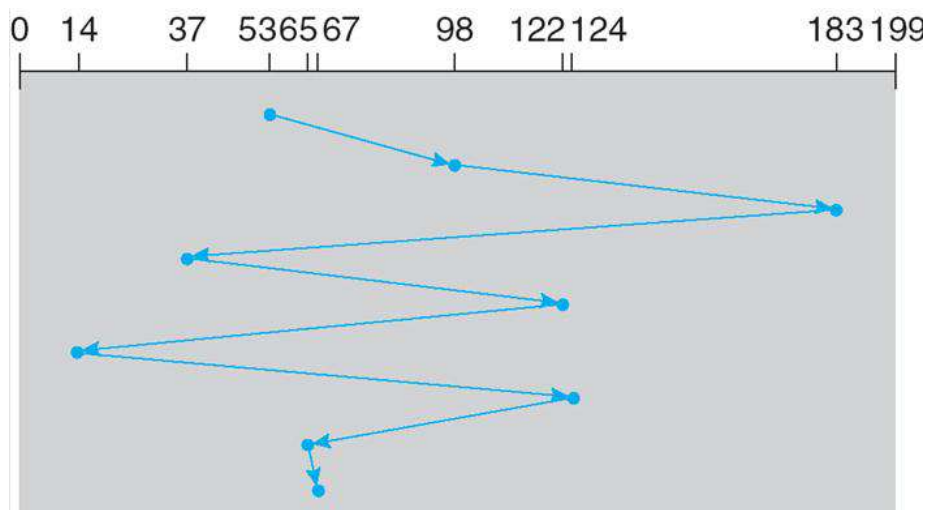
V – Secondary Disk Scheduling Management

There are four main disk scheduling strategies:

- FCFS scheduling
- SSTF scheduling
- SCAN
- C-SCAN

FCFS scheduling

It is the simplest algorithm used in disk scheduling. There is no starvation. This strategy gives service to the request whose localization is at the head of the ready queue. It is good for few processes. But this algorithm is not efficient because its header movement is higher than other algorithm. It's not the fastest service. For example, if we have the following request queue: 98, 183, 37, 122, 14, 124, 65, and 67. Thanks to FCFS, we can obtain this:



As you can see, the header is 53. So we start at 53, then we follow the request queue and go to a point to another: 53->98->183...->65->67. Then, we have to see the number of tracks. To determine the number of tracks, we use the request queue and we subtract a number in the queue by the next one:

Request queue: 53, 98, 183, 37, 122, 14, 124, 65 and 67.



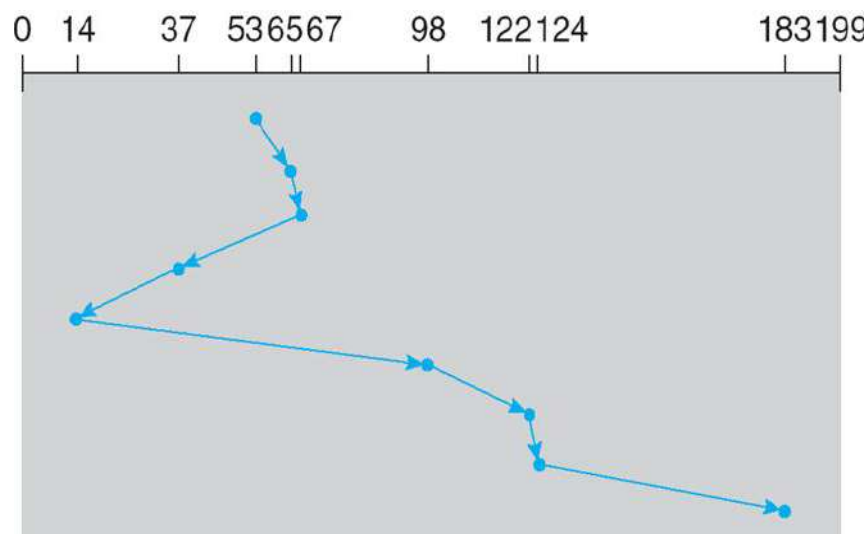
$98 - 53 = 45$
 $183 - 98 = 85$
 $183 - 37 = 146$
 $122 - 37 = 85$
 $122 - 14 = 108$
 $124 - 14 = 110$
 $124 - 65 = 59$
 $67 - 65 = 2$

Then, we make the total:

$45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 = 640 \text{ tracks}$

SSTF scheduling

This algorithm gives a service to request whose position is the nearest to the header. SSTF makes the header move the less possible. It selects the item which requires the shortest seek time. We prefer to use this algorithm than the FCFS one in terms of performance. Let's see:



Request queue: 53, 98, 183, 37, 122, 14, 124, 65 and 67.

The nearest request for 53 is 65. Then the nearest number for 65 is 67, then 37, etc... We always choose the nearest number. We do the same that for the FCFS and we find the total:

$12 + 2 + 30 + 23 + 64 + 24 + 2 + 59 = 236 \text{ tracks}$

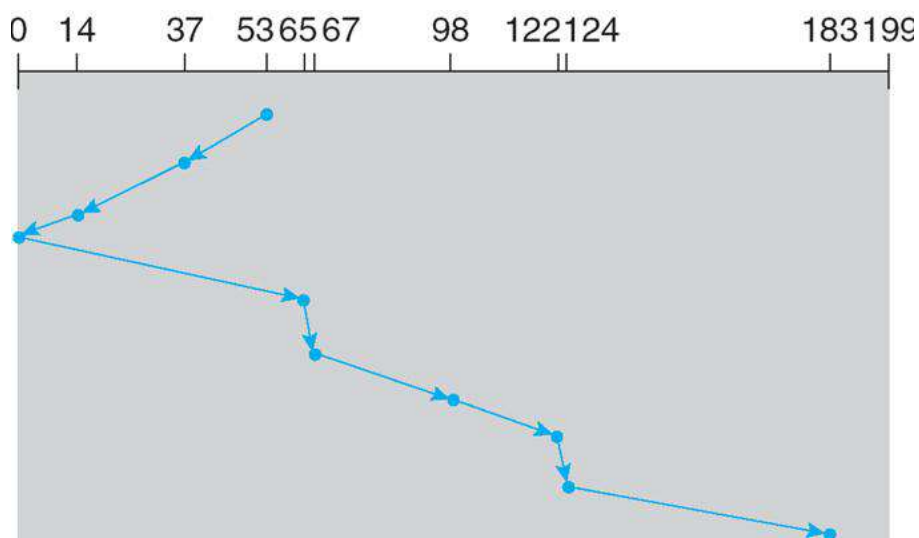


SCAN

We need to have two parameters for the SCAN algorithm:

- **Position** of the header
- **Direction** of the header

The disk arm begins at one end of the disk and moves until the other end. When the last track is reached, the only one direction of heads is reversed. There is no danger of starvation for this algorithm; it is similar to the SSTF scheduling. This algorithm is also called "**elevator algorithm**".

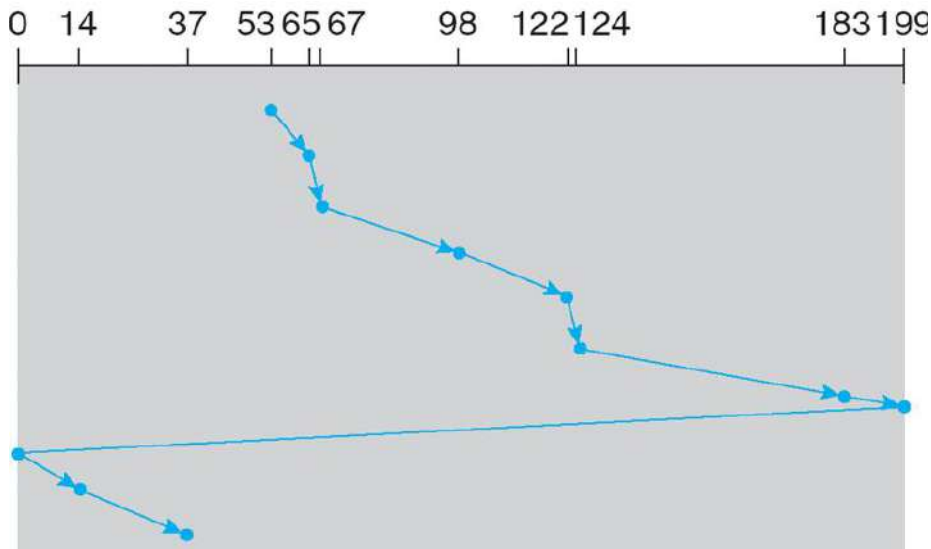


As you can see, it begins at the position 53 (header). Then, the direction of the header is left. It changes direction after reaching 0 and will continue right until the end. We finally find **236 tracks**.



C-SCAN

It provides a uniform wait time. The head moves from one end of the disk to another. It serves requests when it moves. It returns to the beginning of the disk when it reaches the end of the disk and doesn't serve requests when it returns. It reduces the maximum delay for new requests.



After reaching 199, it has to return to the other end (0). Then it changes direction but we see that during its return, it doesn't serve any request. But when it reaches 0, it changes direction and then it can serve requests. We finally find **183 tracks**.

To sum up, we have:

FCFS : 640 tracks

SSTF : 236 tracks

SCAN : 236 tracks

C-SCAN : 183 tracks