History

So the monitor was just a program

History

So the monitor was just a program

It would load an application into memory (from tape or wherever)

History

So the monitor was just a program

It would load an application into memory (from tape or wherever)

And then jump to the start of the application and start executing it

History

So the monitor was just a program

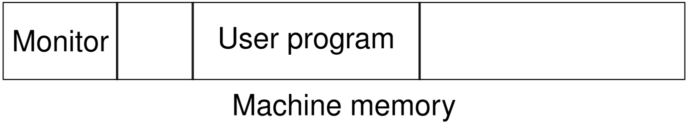
It would load an application into memory (from tape or wherever)

And then jump to the start of the application and start executing it

When the application finished, it would (be expected to) jump back to the monitor, so it could deal with the next program

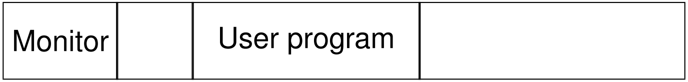
History

But if the application was badly written, it could overwrite the monitor



History

But if the application was badly written, it could overwrite the monitor

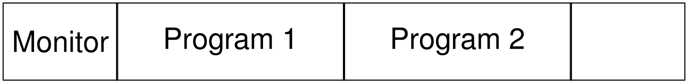


Machine memory

Either accidentally or deliberately

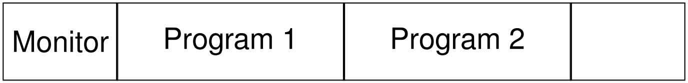
History

It was soon found to be more efficient to load more than one program into memory (when there was space)



History

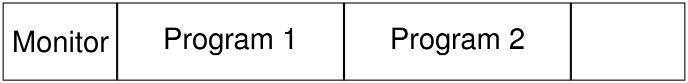
It was soon found to be more efficient to load more than one program into memory (when there was space)



The advantage being that if Program 1 was doing something like writing to a tape that takes a lot of time, but no CPU, the computer could run Program 2 in the meanwhile

History

It was soon found to be more efficient to load more than one program into memory (when there was space)

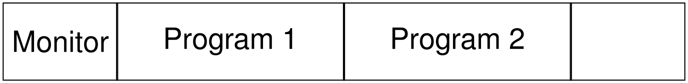


The advantage being that if Program 1 was doing something like writing to a tape that takes a lot of time, but no CPU, the computer could run Program 2 in the meanwhile

When Program 2 pauses and Program 1 needs to run again, the computer could switch back to it

History

It was soon found to be more efficient to load more than one program into memory (when there was space)



The advantage being that if Program 1 was doing something like writing to a tape that takes a lot of time, but no CPU, the computer could run Program 2 in the meanwhile

When Program 2 pauses and Program 1 needs to run again, the computer could switch back to it

The decisions on what to run and actually doing the switching between programs was the job of the monitor

History

Now Program 1 could corrupt Program 2 as well as the monitor!

History

Now Program 1 could corrupt Program 2 as well as the monitor!

Or Program 1 could read confidential data out of Program 2

History

Now Program 1 could corrupt Program 2 as well as the monitor!

Or Program 1 could read confidential data out of Program 2

Some sort of protection of the monitor and other programs is needed

History

Now Program 1 could corrupt Program 2 as well as the monitor!

Or Program 1 could read confidential data out of Program 2

Some sort of protection of the monitor and other programs is needed

What happens if Program 1 goes into an infinite loop?

History

Now Program 1 could corrupt Program 2 as well as the monitor!

Or Program 1 could read confidential data out of Program 2

Some sort of protection of the monitor and other programs is needed

What happens if Program 1 goes into an infinite loop?

Control never returns to the monitor and Program 2 never gets to run

History

Now Program 1 could corrupt Program 2 as well as the monitor!

Or Program 1 could read confidential data out of Program 2

Some sort of protection of the monitor and other programs is needed

What happens if Program 1 goes into an infinite loop?

Control never returns to the monitor and Program 2 never gets to run

Some means of curtailing runaway programs is needed

History

So the “monitor runs the programs”: what does this really mean?

History

So the “monitor runs the programs”: what does this really mean?

Nothing sophisticated. The monitor code just jumps to the program code so the machine is now running the program

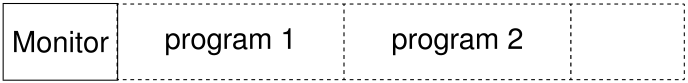
History

So the “monitor runs the programs”: what does this really mean?

Nothing sophisticated. The monitor code just jumps to the program code so the machine is now running the program

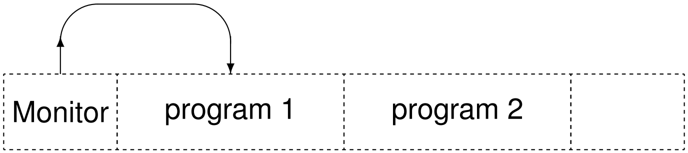
Take care over this point: the monitor doesn’t sit and watch the program running, the monitor is *not* running while the program is running

History



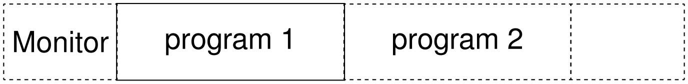
Monitor runs

History



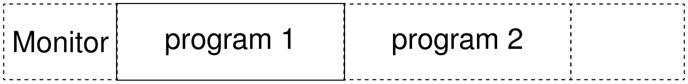
Monitor jumps to program 1

History



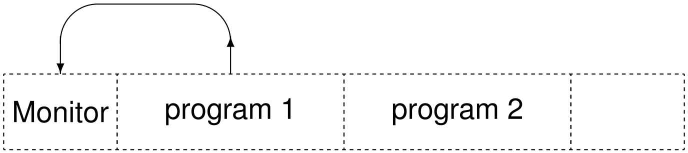
Program runs

History



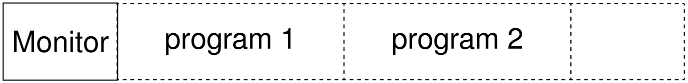
Tape needed

History



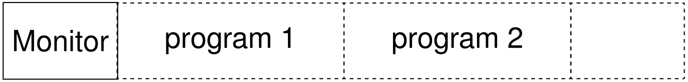
Program calls monitor

History



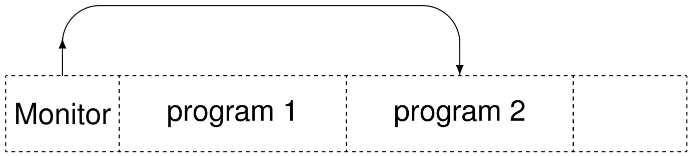
Monitor sets up tape

History



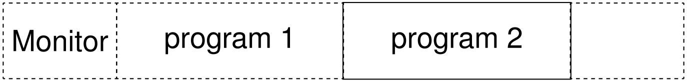
Monitor decides to run another program while waiting for the tape

History



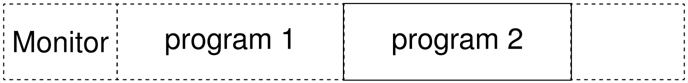
Monitor jumps to program 2

History



Program 2 runs

History



Etc.

History

Less graphically:

History

Less graphically:

Monitor starts program

History

Less graphically:

Monitor starts program $\rightarrow$

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs. . .

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs. . . tape needed $\rightarrow$

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape…

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape… Monitor decides to run another program

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape… Monitor decides to run another program $\rightarrow$

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs... tape needed $\rightarrow$ Monitor sets up tape... Monitor decides to run another program $\rightarrow$ Prog 2

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape… Monitor decides to run another program $\rightarrow$ Prog 2 etc.

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape… Monitor decides to run another program $\rightarrow$ Prog 2 etc.

There is a *single stream of control* jumping between monitor and several programs

History

Less graphically:

Monitor starts program $\rightarrow$ Prog 1 it runs… tape needed $\rightarrow$ Monitor sets up tape… Monitor decides to run another program $\rightarrow$ Prog 2 etc.

There is a *single stream of control* jumping between monitor and several programs

The monitor is not running when a user program is running, and vice-versa

History

This changing between multiple user programs is called *multitasking*. But only one thing is ever running

$\square$

History

This changing between multiple user programs is called *multitasking*. But only one thing is ever running

Multitasking improves the efficiency of use of a computer since while one program waits for a slow peripheral another program can run

□