The history of the C programming language is closely tied to the history of the development of the Unix Operating System.

If we look back to understand what led to the development of the operating system that changed the world of computing,

we'll see the steps that led to the development of C.

Simply put, C was derived from the need to initially find and eventually create a language to apply on the Unix

Operating system.

It all started in 1965 when the experimental project MAC was completed at MIT – the first system of its kind.

This was the beginning of the MULTICS era. It used something called CTSS, or the Compatible Time Sharing System.

This was a key innovation at that time. Up to this point, we were in the early mainframe era, where massive, powerful,

and extremely costly computers used to take up entire rooms.

To get tasks done, programmers would write code by hand. Then they'd punch a deck of paper tape cards that were encoded

with the program written by hand.

They did this by handing the sheets of paper the program was written on to operators who used a key punch machine that

would punch the card's holes and represent the data and instructions on the card.

Then they'd feed the punched cards to a punch card reader connected to the mainframe computer. It then converted the

sequences in the cards holes to digital information. Simple tasks took a long time using this method and only one person

could use each machine at a time.

The idea of time sharing changed everything. Instead of using cards, it attached multiple consoles (which at the time

were mechanical terminals called teletypes) to a main computer. This allowed many people to use the same computer

simultaneously.

Over 100 typewriter terminals spread around MIT's campus could be attached to one main big computer. This system

supported up to 30 remote users at the same time, each using one of those terminals.

The operating system of the main computer multitasked and circled around the people who wanted to perform computing

tasks from their connected terminals and gave a few seconds to each one.

It provided what seemed like a continuous service, appearing to be loading and running many programs simultaneously.

But in reality it just went through each user's program very quickly. This gave the illusion that one person had the

whole computer to themselves.

This system proved to be extremely efficient, effective, and productive, saving time and in the long run money, since

those computers were extremely expensive.

Something that might have taken days to complete now took much less time. And this started enabling greater access to

computing.

Following the success of the CTSS, MIT decided it was time to build upon this system and take the next step. This next

step would be to create a more advanced time sharing system.

