I believe the phrase "ontogeny recapitulates phylogeny" is accurate when developing operating systems. New computer systems are developed similarly to their ancestors. Operating systems have concepts no longer used that could be used in a future application. Large memory, protection hardware, disks, virtual memory for example have similar pattern of development.

Each new mainframe, minicomputer, personal computer, embedded computer, smart card, etc., seems to go through the development process that its ancestor did. The first mainframes were written entirely in assembly language. Compilers and operating systems were written in assembly. Even with the common mainframes using FORTRAN, COBOL, or other HLLs minicomputers were programmed in assembly because of lack of memory. When early PCs (microcomputers) were invented, they also started off as programmed in assembly. Palmtop computers and smart cards started in assembly but quickly moved to HLL.

In the case of operating systems, the first mainframes initially had no protection hardware and no support for multiprogramming, so simple operating systems that handled one manually loaded program at a time were ran. Later adding hardware and operating system support to handle multiple programs simultaneously and then full timesharing capabilities. The software development in these cases was dictated by the technology. Without the available memory HLL were too much to handle, but as the hardware advanced it becomes possible to implement more advanced features. This pattern, or wheel of reincarnation, is likely to continue in computer development.