|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter Passing Implementation Model | Pass-by-Value | Pass-by-Result | pass-by-value-result | pass-by-reference | pass-by-name |
| Description | model for in mode parameters | model for outmode parameters, no value is transmitted to subprogram | inout mode semantics in which actual values are copied | uses a reference of the actual parameter | the actual parameter is textually transmitted to the formal parameter |
| Semantics Model | In mode | outmode | uses inout mode | inout mode | inout mode |
| Conceptual Model | uses copy to initialize the formal parameter | also uses copy typically | also uses copy | uses an address path (alias) | reference, but on steroids |
| Advantages | it is fast for scalars in linkage and time costs | simialr to pass-by-value, fast for scalars | fast for scalars | no costly copying is required (time or space) | none mentioned |
| Disadvantages | additional storage is required for the copy, and size and linkage can be expensive if the parameter is large like an array | similar to pass-by-value, except it also it can be slow if parameters are returned by copy instead of reference, and it can also have parameter collisions. | similar disadvantages to pass-by-value and result because it copies at the beginning and end of the subprogram | access to the parameter will be slower because of indirect memory access. reduces program readability and reliability | hard to implement and inefficient, not widely used, reduces readability and reliability |