УДК 004.382.7

УСКОРЕНИЕ ДИНАМИЧЕСКОГО ПЕРЕРАСПРЕДЕЛЕНИЯ ОПЕРАТИВНОЙ ПАМЯТИ

Ю. Д СТОЛЯРОВ, В. П. ВАСИЛЕВСКИЙ, Э. И. ЯСЮКОВИЧ
Государственное учреждение высшего профессионального образования
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»
Могилев, Беларусь

При выполнении рабочих программ процессор компьютера постоянно использует оперативную память, в которой находится необходимая в данный момент информация. По мере выполнения работы в памяти появляются участки, информация из которых уже не используется. Для эффективной работы в ЭВМ происходит динамическое перераспределение памяти, когда неиспользуемая информация смещается в конец оперативной памяти, а на ее место перемещается необходимая информация, доступ к которой упрощается за счет последовательного доступа без разрывов адресов. При этом процессор прекращает выполнение основной программы, что снижает эффективность работы компьютера. Во время работы ЭВМ всегда имеются такты ожидания, когда процессор не загружен основной программой и можно использовать это время на перераспределение памяти.

Рассмотрим состояние участка памяти ОЗУ в какой-то момент времени. Участки памяти 2 и 5 представляют собой области, информация из которых в настоящий момент уже не нужна (рис.1, а) и их необходимо переместить в конец памяти. Сначала надо сдвинуть область 2 к области 4, потом к области 5, а затем продолжить сдвигать объединенную область вправо до тех пор, пока все свободные области не окажутся в конце справа. При работе программы приходится производить дополнительные действия, чтобы найти информацию по адресу, выданному процессором. Например, при выдаче процессором адреса, попадающего в неиспользуемую область 2. С этой целью между процессором и ОЗУ располагается специальный преобразователь адресов. Сначала в преобразователь загружается адрес левого края первой неиспользуемой области – А, ширина ее – n и адрес конца свободной области – В, после этого процессор считает, что область 2 уже сдвинулась вправо. В действительности, область 2 сдвигается на каждом такте, не связанном с выполнением рабочей программы вправо на одну ячейку и границы области А и Б также перемещаются вправо.

Если выданный процессором адрес меньше, чем адрес А, т. е. соответствующая ячейка уже попала в область 3а, или расположен левее, то адрес не требует корректировки и адрес из процессора поступает прямо в ОЗУ. Если адрес, поступающий от процессора равен или больше А , но меньше В, значит необходимые данные расположены в области 3б и для получения доступа к этому адресу к нему нужно добавить ширину области n (рис.1, б).

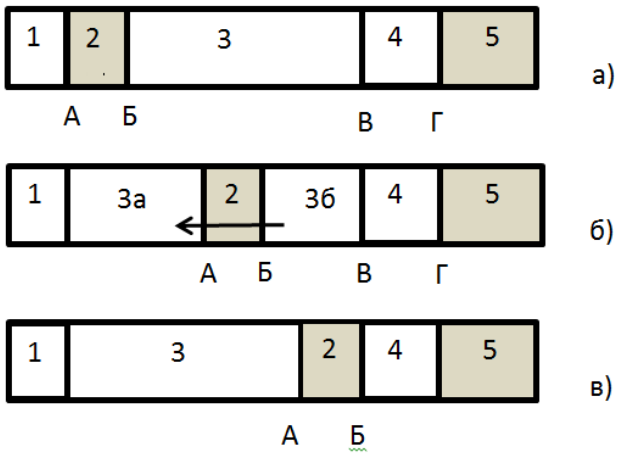

Рис. 1. Последовательность сдвига неиспользуемой части памяти в конец ОЗУ

Пусть А=101, Б=110, т. е. ширина области равна 10, а адрес В=140. Область 3 занимает 30 ячеек памяти. Если процессор выдаст адрес К=101, то т. к. здесь расположена неиспользуемая область памяти 2, информация будет взята по адресу 101+10=111 области 3 с использованием преобразователя адресов. Пусть произошел сдвиг области 2 на 5 ячеек вправо. На это место переместились 5 ячеек области 3 (рис.1, б). Теперь при обращении процессора по адресу К=101 преобразования адреса не требуется, так как информация уже находится в 101 ячейке ОЗУ. Для адреса К=129 информация будет находиться в ячейке ОЗУ с номером 139. После полной замены местами областей памяти 2 и 3, доступ к области 3 рабочей программы будет осуществляться напрямую без преобразования адреса. Если адрес из процессора будет больше адреса В – начала области 4, то преобразования адреса не требуется. Программа, отвечающая за сдвиг области 2, постоянно проверяет условие окончания этого сдвига, при котором разность между текущим адресом А неиспользуемой области 2 и концом области 3 станет равной нулю. Это означает, что область 2 достигла области 4 (рис.1, в). Ширина области 3 может быть больше, меньше или равна n. Далее сдвинем область 2 к неиспользуемой зоне 5. После их объединения продолжим перемещать этот участок вправо, пока все неиспользуемые участки памяти не окажутся в конце, а слева будут расположены адреса рабочих программ, к которым осуществляется последовательный доступ без всяких преобразований. Таким образом, перемещение неиспользуемых областей памяти происходит не за счет времени выполнения рабочей программы, что ускоряет работу компьютера.