

26.26.1 Дати характеристику типових помилок часу виконання в програмах на мові Асемблер.

В оперативній пам'яті більшості комп'ютерних систем в кожний конкретний момент часу може розміщуватись значна кількість різних програм та даних, які розділяють на системні та прикладні. Почергове виконання цих програм в відповідності до потреб експлуатації в загальному випадку може призвести до непередбаченого і несанкціонованого взаємного впливу між ними, який виникає в наслідок помилок в програмах. Як небезпідставно шуткують теоретики програмування, безпомилкова програма - це та програма, в якій помилки ще не виявлені. На рівні машинних команд і відповідно на рівні мови Асемблера несанкціонований вплив між програмами або частинами однієї і тієї програми можна розділити на наступні випадки :

- а) читання даних по помилковій адресі (наприклад, читання байт програми або даних із другої програми), що може привести до неправильної роботи програми, яка читає .
- б) записування даних по помилковій адресі – (наприклад —пошкодження|| команд або

даних другої програми), що призводить до помилок в роботі другої програми, коли вона буде запущена

- с) передача управління по помилковій адресі (як правило, не тільки не на початок якоїсь другої програми, а і не на перший байт команди іншої програми), що призводить до не передбачуваної реакції всієї комп'ютерної системи.

Можна вважати, що непередбачений і несанкціонований взаємний вплив між програмами пов'язаний з помилками в адресації. Інші помилки в програмах приведуть до неправильних результатів її роботи, що може негативно вплинути на інші програми, але такий вплив можна передбачити і нівелювати програмним шляхом.

На рівні мови Асемблера основним джерелом помилок в адресації є помилки в посередній адресації та багатокomпонентній адресації, коли вся адреса або окремі її компоненти розміщуються не в команді, а в регістрах або пам'яті.

Прикладами таких команд, при зверненні до пам'яті, є

Mov	[si],dx
Add	[bx+di],cx
Shl	Mas1[bx],6
Xor	Mas1[bx+si],ax
Push	ax
Pop	dx

Прикладами таких команд при внутрішньо сегментній передачі управління є

Jump	word ptr Mas2[si]
Jump	word ptr [di]
Jump	ax
ret	

Вміст регістрів, які використовуються для адресації, часто буває результатом досить складних обчислень, що є потенційним джерелом помилок. Практика програмування показує, що особливо часто виникають помилки при використанні команди **ret** через те,

що при **виконанні** процедури виник дисбаланс читання та записування в стек . В результаті замість повернення із процедури відбувається передача управління по випадковій адресі.

При сегментній організації пам'яті додатковим джерелом помилок адресації є помилки в завантаженні та використанні сегментних регістрів.