Copying Bytes

M. Anton Ertl, TU Wien

Myths

• Copying bytes efficiently is simple
• Cmove is faster than move
• Implementing cmove efficiently is simple
• Implementing move efficiently is more complex

Cycles for 50-byte non-overlapping copy

<table>
<thead>
<tr>
<th>Skylake sf</th>
<th>sf</th>
<th>gforth</th>
<th>vfx32</th>
<th>vfx64</th>
<th>move</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>36</td>
<td>34</td>
<td>24</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>87</td>
<td>32</td>
<td>21</td>
<td>27</td>
<td>cmove</td>
</tr>
<tr>
<td>83</td>
<td>90</td>
<td>33</td>
<td>21</td>
<td>224</td>
<td>cmove&gt;</td>
</tr>
</tbody>
</table>

byte loop memmove() cell loop rep movsb
Words and C functions

<table>
<thead>
<tr>
<th>Forth</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td>memmove()</td>
</tr>
<tr>
<td>cmove</td>
<td></td>
</tr>
<tr>
<td>cmove&gt;</td>
<td>memcpy()</td>
</tr>
<tr>
<td>move&lt;</td>
<td></td>
</tr>
<tr>
<td>move&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Efficient implementations

: move ( from to u -- )
over 3 pick - 2 pick u< if \ to in [from,from+u)
  move>
else
  move<
then ;

: cmove ( afrom ato u -- )
dup 0= if exit then
begin ( afrom1 ato1 u1 )
  over 3 pick - 2>r
  2dup 2r0 umin move<
  2r0 1 rot within while
  2r> /string repeat
  2r> 2drop 2drop ;

Extend 2-byte pattern to 1000 bytes with cmove

Zen 3 cycles/cmove
VFX64   VFX32
rep movsb  cell loop
orig new orig new
3360  965  4273  386
Conclusion

- Moving bytes efficiently is simple
- Cmove is faster than move? Sometimes
- Implementing cmove efficiently is simple
- Implementing move efficiently is more complex