```
>LIST
   10REM > RISCOSmark 1.01 (14 May 2003)
   30REM by Richard Spencer 2003 released as Hackware
   40REM Here's the source, no license restrictions: have fun!
   50REM Please email suggestions, problems, updates to
   60REM richardspencer@freeuk.com
   70
   71OSCLI("If ""<RISCOSmark$File>""=""""" Then Set RISCOSmark$File @.
~ROmarkF~")
  110
  120END=1280*1024
  130codemax%=64*1024
  140DIM code% codemax%, block% 1024*1024
  150
  160SYS "OS_ReadMonotonicTime" TO t%
  170logfile$="Pipe:$.mlog"+RIGHT$("0000"+STR$(t% MOD 100000),5)
  180log%=OPENOUT(logfile$)
  190ON ERROR ON ERROR OFF:PRINT"Error (see
logfile)":BPUT#log%,REPORT$+" in line "+STR$ERL:CLOSE#log%:OSCLI
("SetType "+logfile$+" FFF"):OSCLI("Filer_Run "+logfile$):END
  200BPUT#log%,"RISCOSmark 1.01 (14 May 2003)"
  210BPUT#log%,"Comparison with RiscPC SA 202MHz running RISC OS 4.02
800x600,256"
  211BPUT#log%,"(HD benchmarks are in kilobytes/sec)"
  220BPUT#log%,10
  230BPUT#log%,"OS/Machine/Processor: ??"
  240
  250VDU 26,20,12:MOUSE OFF
  260SYS "OS_ReadModeVariable",-1,11 TO ,,xpix%
  270xpix%+=1
  280SYS "OS_ReadModeVariable",-1,12 TO ,,ypix%
  290ypix%+=1
  300SYS "OS_ReadModeVariable",-1,4 TO ,,xf%
  310xf%=1<<xf%
  320SYS "OS_ReadModeVariable",-1,5 TO ,,yf%
  330yf%=1<<yf%
  340SYS "OS_ReadModeVariable",-1,3 TO ,,col%
  350CASE col% OF
  360  WHEN 1:col$="2"
  370  WHEN 3:col$="4"
  380  WHEN 15:col$="16"
  390  WHEN 63,255:col$="256"
  400  WHEN 65535:col$="32K"
  410  WHEN -1:col$="16M"
  420  OTHERWISE
  430  col$="unknown"
  440ENDCASE
  450BPUT#log%,"Graphics Resoloution: "+STR$xpix%+"x"+STR$ypix%+",
"+col$+" colours"
  460BPUT#log%,10
  470BPUT#log%,LEFT$("Test"+STRING$(48," "),48)+"       Benchmark"
  480RESTORE +0
  490test%=0
  500READ test$,description$,baseline$,routine$
  510WHILE routine$<>"*"
  520  CLS
  530  test%+=1
  540  PRINT"Running test ";test%;" (";test$;")"
  550  mark%=EVAL("FN"+routine$)
  560  baseline%=VAL(baseline$)
  570  IF baseline% THEN mark$=RIGHT$("       "+STR$(INT
(mark%/baseline%*100))+"%",9) ELSE mark$=""
  580  BPUT#log%,LEFT$(test$+" - "+description$+STRING$(64," "),48)
+RIGHT$(STRING$(16," ")+STR$mark%,16)+mark$
```

```
  590  READ test$,description$,baseline$,routine$
  600ENDWHILE
  610CLOSE #log%
  620OSCLI("SetType "+logfile$+" FFF")
  630OSCLI("Filer_Run "+logfile$)
  631OSCLI("If ""<RISCOSmark$Dir>""=""@.~ROmarkF~"" Then Unset
RISCOSmark$File")
  660PRINT"Tests complete."
  670END
  680
  690DATA Processor,Looped instructions (cache),177868,processor
  700DATA Memory,Multiple register transfer,162,memory
  710DATA Rectangle Copy,Graphics acceleration test,242,rectangle
  720DATA Icon Plotting,16 colour sprite with mask,2000,icon
  730DATA Draw Path,Stroke narrow line,1560,drawpath
  740DATA Draw Fill,Plot filled shape,1459,drawfill
  750DATA HD Read,Block load 1MB file,2982,loadblock
  760DATA HD Write,Block save 1MB file,3041,saveblock
  770DATA FS Read,Byte stream file in,207,streamin
  780DATA FS Write,Byte stream file out,192,streamout
  790DATA ,,,*
  800
  810DEF FNentry
  820[OPT pass%
  830.timeupflag%    EQUD    0
  840.timeup%        STR     r12,[r12] ; store nonzero flag to
  850                MOV     pc,r14    ; indicate time up
  860.entry%         SWI     "OS_ReadMonotonicTime"
  870                MOV     r1,r0
  880.tickl%         SWI     "OS_ReadMonotonicTime"
  890                CMP     r1,r0
  900                BEQ     tickl%  ; wait for tick
  910                MOV     r0,#100
  920                ADR     r1,timeup%
  930                ADR     r2,timeupflag%
  940                SWI     "OS_CallAfter"
  950                MOV     r0,#0   ; benchmark return register
  960                ADR     r12,timeupflag%
  970]=0
  980
  990DEF FNprocessor
 1000FOR pass%=8 TO 10 STEP 2
 1010  P%=code%
 1020  L%=code%+codemax%
 1030[OPT pass%
 1040.tempstr%      EQUD    0
 1050               FNentry
 1060               MOV     r10,#0
 1070               ADR     r4,tempstr%
 1080.processorl% ]
 1090  FOR instrl%=1 TO 80 : REM 12*4*80=3840 fit in Arm7/SA cache
 1100[OPT pass%
 1110               MOV     r1,#123<<13
 1120               MOV     r2,#234<<11
 1130               ADD     r1,r1,#167<<6
 1140               ADD     r2,r2,#121<<4
 1150               LDR     r3,[r4] ; test data cache a bit
 1160               SUB     r3,r1,r2
 1170               ADD     r3,r1,r2
 1180               AND     r3,r1,r2
 1190               ORR     r3,r1,r2
 1200               EOR     r3,r1,r2
 1210               MUL     r3,r1,r2
 1220               STR     r3,[r4]
```

```
1230] NEXT
1240[OPT pass%
1250                ADD     r0,r0,#1
1260                LDR     r11,[r12]
1270                CMP     r11,#0
1280                BEQ     processorl%
1290                MOV     pc,r14
1300]NEXT
1310=USR(entry%)
1320
1330DEF FNmemory
1340FOR pass%=8 TO 10 STEP 2
1350  P%=code%
1360  L%=code%+codemax%
1370[OPT pass%
1380.datast%        EQUD    block%
1390.dataen%        EQUD    block%+1024*1024
1400                FNentry
1410                MOV     r11,#0
1420.memoryl%       ADR     r9,datast%
1430                LDR     r9,[r9]
1440                ADR     r10,dataen%
1450                LDR     r10,[r10]
1460.memoryl2% ]
1470  FOR meminstrl%=1 TO 256 : REM NB <=2048
1480[OPT pass%
1490                LDMIA   r9!,{r1-r8}
1500                STMDB   r10!,{r1-r8}
1510] NEXT
1520[OPT pass%
1530                ADD     r11,r11,#1
1540                CMP     r11,#16
1550                BNE     memoryl2%
1560                MOV     r11,#0
1570                ADD     r0,r0,#1
1580                LDR     r11,[r12]
1590                CMP     r11,#0
1600                BEQ     memoryl%
1610                MOV     pc,r14
1620]NEXT
1630=USR(entry%)
1640
1650DEF FNrectangle
1660rectx%=301
1670recty%=237
1680IF rectx%>xpix% OR recty%>ypix% THEN =0
1690SYS "Wimp_SetColour",9
1700RECTANGLE FILL 0,0,rectx%*xf%,recty%*yf%
1710SYS "Wimp_SetColour",11
1720RECTANGLE FILL 8*xf%,8*yf%,(rectx%-16)*xf%,(recty%-16)*yf%
1730FOR i%=0 TO 1023
1740  block%!(i%*8)=RND((xpix%-rectx%)*xf%)
1750  block%!(i%*8+4)=RND((ypix%-recty%-8)*yf%)
1760NEXT
1770FOR pass%=8 TO 10 STEP 2
1780  P%=code%
1790  L%=code%+codemax%
1800[OPT pass%
1810.blockaddr%     EQUD    block%
1820                FNentry
1830                MOV     r11,#0
1840                MOV     r10,#0 ; index for reading rect pos
1850                ADR     r9,blockaddr%
1860                LDR     r9,[r9]
```

```
1870                MOV     r1,#0 ; new x and y for rect
1880                MOV     r2,#0
1890.rectl%         MOV     r0,#4
1900                SWI     "OS_Plot" ; move to b l corner
1910                ADD     r1,r1,#(rectx%*xf%) AND 255
1920                ADD     r2,r2,#(recty%*yf%) AND 255
1930                ADD     r1,r1,#(rectx%*xf%) AND 255<<8
1940                ADD     r2,r2,#(recty%*yf%) AND 255<<8
1950                SWI     "OS_Plot" ; move to t r corner
1960                LDR     r1,[r9,r10,LSL #2]
1970                ADD     r10,r10,#1
1980                LDR     r2,[r9,r10,LSL #2]
1990                ADD     r10,r10,#1
2000                CMP     r10,#256
2010                MOVEQ   r10,#0
2020                MOV     r0,#190 ; copy to new b l
2030                SWI     "OS_Plot"
2040                ADD     r11,r11,#1
2050                LDR     r0,[r12]
2060                CMP     r0,#0
2070                BEQ     rectl%
2080                MOV     r0,r11
2090                MOV     pc,r14
2100]NEXT
2110=USR(entry%)
2120
2130DEF FNicon
2140spr%=block%
2150sc%=block%+16*1024
2160tr%=block%+20*1024
2170xy%=block%+32*1024
2180!spr%=16*1024
2190spr%!4=0
2200spr%!8=16
2210spr%!12=16
2220SYS "OS_SpriteOp",9+256,spr%
2230SYS "OS_SpriteOp",15+256,spr%,"icon",,64,64,27
2240SYS "OS_SpriteOp",29+256,spr%,"icon"
2250SYS "OS_SpriteOp",61+256,spr%,"icon",0 TO a,b,c,d
2260GCOL 0,0
2270RECTANGLE FILL 0,0,128,128
2280GCOL 0,15
2290CIRCLE FILL 64,64,64
2300SYS "OS_SpriteOp",a,b,c,d
2310SYS "OS_SpriteOp",60+256,spr%,"icon",0 TO a,b,c,d
2320GCOL 0,9
2330CIRCLE FILL 64,64,64
2340GCOL 0,11
2350CIRCLE FILL 64,64,48
2360SYS "OS_SpriteOp",a,b,c,d
2370SYS "Wimp_ReadPixTrans",256,spr%,"icon",,,,sc%,tr%
2380FOR i%=0 TO 1023
2390  xy%!(i%*8)=RND((xpix%-64)*xf%)
2400  xy%!(i%*8+4)=RND((ypix%-64-8)*yf%)
2410NEXT
2420FOR pass%=8 TO 10 STEP 2
2430  P%=code%
2440  L%=code%+codemax%
2450[OPT pass%
2460.spraddr%       EQUD    spr%
2470.scaddr%        EQUD    sc%
2480.traddr%        EQUD    tr%
2490.xyaddr%        EQUD    xy%
2500                FNentry
```

```
 2510                MOV     r11,#0
 2520                MOV     r10,#0 ; index for reading rect pos
 2530                ADR     r9,xyaddr%
 2540                LDR     r9,[r9]
 2550                MOV     r0,#52 ; OS_SpriteOp 512+52
 2560                ADD     r0,r0,#512
 2570                ADR     r1,spraddr%
 2580                LDR     r1,[r1]
 2590                ADD     r2,r1,#16
 2600                ADR     r6,scaddr%
 2610                LDR     r6,[r6]
 2620                ADR     r7,traddr%
 2630                LDR     r7,[r7]
 2640                MOV     r5,#8
 2650.iconl%         LDR     r3,[r9,r10,LSL #2]
 2660                ADD     r10,r10,#1
 2670                LDR     r4,[r9,r10,LSL #2]
 2680                ADD     r10,r10,#1
 2690                CMP     r10,#1024
 2700                MOVEQ   r10,#0
 2710                SWI     "OS_SpriteOp"
 2720                ADD     r11,r11,#1
 2730                LDR     r8,[r12]
 2740                CMP     r8,#0
 2750                BEQ     iconl%
 2760                MOV     r0,r11
 2770                MOV     pc,r14
 2780]NEXT
 2790=USR(entry%)
 2800
 2810DEF FNdrawpath
 2820LOCAL DATA
 2830RESTORE +0
 2840DATA 2,3,6,6,2,237*yf%<<8,500*xf%<<8,266*yf%<<8,101*xf%<<8,93
*yf%<<8,0,&80000000
 2850i%=block%
 2860READ d%
 2870WHILE d%<>&80000000
 2880  !i%=d%
 2890  i%+=4
 2900  READ d%
 2910ENDWHILE
 2920SYS "ColourTrans_SetGCOL",&F3774200,,,256,0
 2930FOR pass%=8 TO 10 STEP 2
 2940  P%=code%
 2950  L%=code%+codemax%
 2960[OPT pass%
 2970.blockaddr%     EQUD    block%
 2980                FNentry
 2990                MOV     r1,#0
 3000                MOV     r2,#0
 3010                MOV     r3,#0
 3020                MOV     r4,#0
 3030                MOV     r5,#0
 3040                MOV     r6,#0
 3050                MOV     r11,#0
 3060.strokel%       ADR     r0,blockaddr%
 3070                LDR     r0,[r0]
 3080                SWI     "Draw_Stroke"
 3090                ADD     r11,r11,#1
 3100                LDR     r10,[r12]
 3110                CMP     r10,#0
 3120                BEQ     strokel%
 3130                MOV     r0,r11
```

```
3140                MOV     pc,r14
3150]NEXT
3160=USR(entry%)
3170
3180DEF FNdrawfill
3190LOCAL DATA
3200RESTORE +0
3210DATA 2,3,6,6,2,237*yf%<<8,500*xf%<<8,266*yf%<<8,101*xf%<<8,93
*yf%<<8,0,&80000000
3220i%=block%
3230READ d%
3240WHILE d%<>&80000000
3250  !i%=d%
3260  i%+=4
3270  READ d%
3280ENDWHILE
3290SYS "ColourTrans_SetGCOL",&F3774200,,,256,0
3300FOR pass%=8 TO 10 STEP 2
3310  P%=code%
3320  L%=code%+codemax%
3330[OPT pass%
3340.blockaddr%     EQUD    block%
3350                FNentry
3360                MOV     r1,#0
3370                MOV     r2,#0
3380                MOV     r3,#0
3390                MOV     r11,#0
3400.filll%         ADR     r0,blockaddr%
3410                LDR     r0,[r0]
3420                SWI     "Draw_Fill"
3430                ADD     r11,r11,#1
3440                LDR     r10,[r12]
3450                CMP     r10,#0
3460                BEQ     filll%
3470                MOV     r0,r11
3480                MOV     pc,r14
3490]NEXT
3500=USR(entry%)
3510
3520DEF FNloadblock
3530readtime%=&42
3540osfile%=8
3550SYS readtime% TO t%
3560c%=0
3570SYS osfile%,11,"<RISCOSmark$File>",&FFD,,0,1024*1024
3580REPEAT
3590SYS osfile%,16,"<RISCOSmark$File>",block%
3600c%+=1
3610SYS readtime% TO t2%
3620UNTIL t2%>t%+100
3630SYS osfile%,6,"<RISCOSmark$File>"
3640=c%*1024*100/(t2%-t%)
3650
3660DEF FNsaveblock
3670readtime%=&42
3680osfile%=8
3690SYS readtime% TO t%
3700c%=0
3710REPEAT
3720SYS osfile%,10,"<RISCOSmark$File>",&FFD,,block%,block%+1024*1024
3730c%+=1
3740SYS readtime% TO t2%
3750UNTIL t2%>t%+100
3760SYS osfile%,6,"<RISCOSmark$File>"
```

```
3770=c%*1024*100/(t2%-t%)
3780
3790DEF FNstreamin
3800SYS "OS_File",11,"<RISCOSmark$File>",&FFD,,0,1024*1024
3810SYS "OS_Find",&4F,"<RISCOSmark$File>" TO stream%
3820FOR pass%=8 TO 10 STEP 2
3830  P%=code%
3840  L%=code%+codemax%
3850[OPT pass%
3860.streamval%     EQUD    stream%
3870                FNentry
3880                STMFD   r13!,{r14}
3890                MOV     r10,#0
3900                ADR     r1,streamval%
3910                LDR     r1,[r1]
3920.streaminl%     SWI     "OS_BGet"
3930                BLCS    resetptr%
3940                ADD     r10,r10,#1
3950                LDR     r11,[r12]
3960                CMP     r11,#0
3970                BEQ     streaminl%
3980                MOV     r0,r10
3990                LDMFD   r13!,{pc}
4000
4010.resetptr%      MOV     r0,#1
4020                MOV     r2,#0
4030                SWI     "OS_Args"
4040                MOV     pc,r14
4050]NEXT
4060m%=USR(entry%)
4070SYS "OS_Find",0,stream%
4080SYS "OS_File",6,"<RISCOSmark$File>"
4090=m%/1024
4091
4100DEF FNstreamout
4120SYS "OS_Find",&8F,"<RISCOSmark$File>" TO stream%
4130FOR pass%=8 TO 10 STEP 2
4140  P%=code%
4150  L%=code%+codemax%
4160[OPT pass%
4170.streamval%     EQUD    stream%
4180                FNentry
4200                MOV     r10,#0
4210                ADR     r1,streamval%
4220                LDR     r1,[r1]
4221                MOV     r0,#&77
4230.streamoutl%    SWI     "OS_BPut"
4250                ADD     r10,r10,#1
4260                LDR     r11,[r12]
4270                CMP     r11,#0
4280                BEQ     streamoutl%
4290                MOV     r0,r10
4350                MOV     pc,r14
4360]NEXT
4370m%=USR(entry%)
4380SYS "OS_Find",0,stream%
4390SYS "OS_File",6,"<RISCOSmark$File>"
4400=m%/1024
>*SPOOL
```