

]LIST

```
1 PRINT CHR$(4)"MAXFILES 1": PRINT CHR$(4)"BRUN TUTMC":UB = PEEK (115) +
256 * PEEK (116): CALL UB: DEFINT I-N,X: DIM I0(127),I1(127),I2(127),XH(2),XV(2)
2 PRINT CHR$(4)"PR#0": TEXT : HOME :XC = 0: ONERR GOTO 4000
3 VTAB 1: HTAB 15: PRINT "TUTORIAL 2": POKE 795,0
4 XH(0) = 1:XV(0) = 2:XH(1) = 1:XV(1) = 21:XH(2) = 1:XV(2) = 6: DEF FN DR(A) =
PEEK (A) + 256 * PEEK (A + 1): DEF FN AR(CX) = INT (CX * 100) * .01
5 RE = FN DR(40160): POKE 40157,4: REM # OF APPLESOFT INSTRUCTIONS RUN BETWEEN
TWO SWITCHES
6 CALL RE,4,IT,I0,2,0,0,1000,I1,1,780,774,2000,I2,2,774,768,3000
7 CALL RE,11,1,1,"PROGRAM ENDED, PRESS ANY KEY";: GET A$: HOME : END
999 REM FIRST COROUTINE: MONITOR EVERY CONTEXT INCLUDING ITSELF
1000 AD = FN DR(40152):OF = 0:NT = 0:SO = PEEK (AD + 17):SL = 0:LX = - 1: GO
SUB 5010: PRINT " RUNNING TASKS STATUS (";SO"/";: GOSUB 5000:XH = XH(IT):XV = X
V(IT)
1002 FOR JT = 0 TO 7: ON PEEK (AD + 8 + JT) < 255 GOTO 1003:NT = JT - 1:JT =
7
1003 NEXT JT: FOR J0 = 0 TO 1 STEP 0: GOSUB 1100:JF = 1
1004 FOR JT = 0 TO NT: GOSUB 1200: NEXT JT
1005 J0 = JF: NEXT J0: RETURN
1099 REM
1100 ON PEEK (40157) = LX GOTO 1102: POKE 40154,128: HTAB XH: VTAB XV:LX = P
EEK (40157)
1101 PRINT LX;"");: CALL - 868: POKE 40155,1: POKE 40154,0
1102 RETURN
1199 REM PRINT A CONTEXT CONTENT (JT)
1200 IF PEEK (AD + JT + 8) < 255 AND JT < > IT THEN JF = 0
1201 OF = PEEK (AD + JT + 8) * 256 + PEEK (AD + JT):XV(IT) = 3 + JT:XH(IT) =
1: GOSUB 5010: CALL 777,OF,JT: GOSUB 5000
1202 RETURN
1999 REM SECOND CONTEXT: PROCESS SOME KEYBOARD INPUT FROM USER
2000 CALL 786:BS$ = CHR$(8):CU$ = CHR$(127) + BS$: POKE 49168,0: ONERR GO
TO 2900
2001 GOSUB 5010: PRINT SPC( 6);"DIVISION EXEMPLE": GOSUB 5000:LY = XV(IT): FO
R J1 = 0 TO 1 STEP 0
2002 XH(IT) = 1:XV(IT) = LY: GOSUB 5010: CALL - 958: PRINT "ENTER NUMERATOR: "
CU$;: GOSUB 2801: ON M$ = "" GOTO 2004:VN = VAL (M$)
2003 GOSUB 5010: PRINT "ENTER DIVISOR: "CU$;: GOSUB 2801: ON M$ < > "" GOTO 2
005
2004 J1 = 1
2005 ON J1 = 1 GOTO 2007:VD = VAL (M$):VR = FN AR(VN / VD): GOSUB 5010: PRIN
T "RESULT: ";VR;" <RET> TO PROCEED"CU$;: GOSUB 2851: ON XC = 1 OR ES% = 1 GOTO 2
004: GOTO 2007
2006 POKE 40154,128: CALL RE,11,24,1,MO$;
2007 NEXT : RETURN
2800 REM INPUT SUBROUTINE
2801 GOSUB 5000:M$ = "":LM = 0:ES% = 0: FOR JS = 0 TO 1 STEP 0
2802 GOSUB 2861: ON ES% = 0 AND XC = 0 GOTO 2803:M$ = "":LM = 0: GOTO 2809
2803 ON JS = 1 GOTO 2809: ON A < > 8 OR LM = 0 GOTO 2804:LM += 1:M$ = LEFT$
(M$,LM + ( @ = 0)): PRINT " A$A$;CU$;: ON LM > 0 GOTO 2804:M$ = ""
2804 ON A < 31 GOTO 2809:LM += 1:M$ += A$: PRINT A$;CU$;
2809 GOSUB 5000: NEXT
2810 GOSUB 5010: CALL - 868: PRINT : GOSUB 5000: RETURN
2850 REM GET RETURN SUBROUTINE
2851 GOSUB 5000:ES% = 0: FOR JS = 0 TO 1 STEP 0
2852 GOSUB 2861:JS = (ES% = 1) OR (XC = 1) OR (A = 13): GOSUB 5000: NEXT : GOS
UB 5010: CALL - 868: GOSUB 5000: RETURN
2860 REM GET KEYBOARD ENTRY
2861 CALL 783: ON PEEK (796) < 128 GOTO 2861
2862 GOSUB 5010: IF XC = 0 THEN GET A$:A = ASC (A$)
2863 ON XC = 0 GOTO 2864: PRINT "#ABORTED#!";:JS = 1
2864 ON A < > 27 GOTO 2865: PRINT "<ESCAPED>";:JS = 1:ES% = 1
2865 ON A < > 13 GOTO 2866:JS = 1
```

```

2866 RETURN
2900 ON PEEK (222) < > 255 GOTO 2902
2901 XC = 1:A$ = CHR$ (3):A = 3: PRINT CHR$ (7);: RESUME
2902 ON PEEK (222) < > 133 GOTO 2903:EL = FN DR(218): ON EL < > 2005 GOTO
2903:MO$ = "DIVIDE BY ZERO ERROR":J1 = 1: CALL - 3288: GOTO 2006
2903 PRINT CHR$ (7);: GOTO 4003
2998 REM 3RD CONTEXT MAIN ROUTINE, JUST PRINT SOME STAR CHARACTERS
2999 REM AS A BACKGROUND ACTIVITY
3000 FOR J2 = 0 TO 1 STEP 0:J2 = J1: GOSUB 3008
3001 PRINT "*";: GOSUB 5000: NEXT
3002 FOR J2 = 1 TO 4: GOSUB 3008: PRINT MID$ ("OVER",J2,1);: GOSUB 5000: NEXT
: RETURN
3008 XV(IT) = INT ( RND (1) * 15) + 6:XH(IT) = INT ( RND (1) * 40) + 1: GOSUB
5010: RETURN
4000 IF PEEK (40156) < 128 THEN VTAB 23: HTAB 1: CALL 771: END
4001 ON PEEK (222) = 255 GOTO 2901
4003 XH(IT) = 1:XV(IT) = 23: GOSUB 5010: CALL 771: GOSUB 5000: CALL RE,5
5000 CALL 789: RETURN : REM STORE CURSOR POSITION INTO CONTEXT AND RELEASE CON
TROL TO MT, EXPECTS TO BE CALLED WHILE CONTEXT SWITCHES INHIBITED
5010 CALL 792: RETURN : REM INHIBIT CONTEXT SWITCH AND RESTORE CURSOR POSITION
FROM STORED CONTEXT

```

]PR#0