

Grep for Sparta DOS#

General Information

Author: Carsten Strotmann

Language: ACTION!

Compiler/Interpreter: ACTION!

Published: 2003

Download: [Grep for Sparta DOS/grep.atr](#)

GREP tool for Sparta DOS#

Table of Contents

- [Grep for Sparta DOS](#)
- [GREP tool for Sparta DOS](#)
- [Limitations:](#)
- [To-Do:](#)
- [Usage:](#)
- [Source \(ACTION!\)](#)

This is the first, very limited alpha version.

Limitations:#

- Some bugs
- only works on Sparta DOS 32GX
- only works with ACTION! Cart
- don't exit to DOS
- only case sensitive
- no parameter checking

To-Do:#

- Case insensitive search
- parameter checking
- replace PEEK with BYTE POINTER
- catch all bugs
- trim RUNTIME part
- use generic Sparta DOS Library

Usage:#

```
grep filename searchstring
```

Source (ACTION!)#

```
*****  
**
```

```
;** SPARTA DOS GREP **
;**
;** (C) 2003 / GPL **
;** CARSTEN STROTMANN **
;**
;*****
;
```

```
;SET 14=$4000
;SET $491=$4000 ; ORG = $4000
```

```
MODULE ; SYS.ACT
```

```
DEFINE EOL="$9B"
DEFINE OpenBuf = "$0500"
DEFINE OpenBufL = "$00"
DEFINE OpenBufH = "$05"
```

```
BYTE ARRAY copy_right(0) =
    "(c) PSC-ACTION! (parts by A.C.S)"
```

```
; Primitive IO routines
PROC Clos=(BYTE d)[$FFA2$A686$CA0$AD0]
```

```
PROC Outputq=(BYTE d,BYTE ARRAY s)
[$A684$BA0$4D0]
```

```
PROC In=(BYTE d,BYTE ARRAY s)
[$A684$5A0$A586$A2$0$A386]
```

```
PROC XIOstr=(BYTE d,x,c,a1,a2,BYTE ARRAY s)
[$A0A$A0A$98AA$9D$342$A3A5$AF0$9D$34A$A4A5$9D$34B$A9$0$9DA8$349
$A5B1$9D$348$12F0$18$A5A5$169$9D$344$A6A5$69$0$9D$345$4C$E456$60]
```

```
PROC Opn=(BYTE d,BYTE ARRAY s,BYTE m,o)
[$A586$A684$3A0$4CXIOstr]
```

```
PROC Prt=(BYTE d,BYTE ARRAY s)
[$A586$A684$A2$0$A386$9A0$20XIOstr$AD0$BA9$9D$342$9BA9$4C$E456$60]
```

```
PROC Error(BYTE err)[$6C$A$0$1113$8301]
```

```
PROC Break=*( )
[$BA$8E$4C1$80A0$98$4C Error]
```

```
; math library routines
PROC LShift=*( )
[$84A4$AF0$8586$A$8526$88$FAD0$85A6$60]
```

```
PROC RShift=*( )
[$84A4$AF0$8586$8546$6A$88$FAD0$85A6$60]
```

```
PROC SetSign=*( )[$D3A4$1010]
PROC SS1=*( )
[$8685$8786$38$A9$0$86E5$A8$A9$0$87E5$AA$98$60]
```

```
PROC SMOps=*( )
```

```
[$D386$E0$0$310$20SS1$8285$8386$85A5$E10$AA$D345$D385
$84A5$20SS1$8485$8586$A9$0$8785$60]
```

```
PROC MultB=*()
[$1BF0$CA$C786$AA$15F0$C686$A9$0$8A2$A$C606$290$C765$CA$F6D0
$18$8765$8785$86A5$87A6$60]
```

```
PROC MultI=*()
[$20SMOps$82A6$1BF0$C686$84A6$15F0$CA$C786
$8A2$A$8726$C606$690$C765$290$87E6$CA$F0D0
$8685$82A5$85A6$20MultB$83A5$84A6$20MultB$4CSetSign]
```

```
PROC DivI=*()
[$20SMOps$85A5$27F0
$8A2$8226$8326$8726$38$83A5$84E5$A8$87A5$85E5$490
$8785$8384$CA$E7D0$82A5$2A$A2$0$83A4$8684$4CSetSign
$10A2$8226$8326$2A$4B0$84C5$390$84E5$38$CA$EFD0
$8226$8326$8685$82A5$83A6$4CSetSign]
```

```
PROC RemI=*()[$20 DivI$86A5$87A6$60]
```

```
PROC SArgs=*()
[$A085$A186$A284$18$68$8485$369$A8$68$8585$69$0$48$98$48$1A0
$84B1$8285$C8$84B1$8385$C8$84B1
$A8$B9$A0$0$8291$88$F810$11A5$FD0$11E6$4C Break$6308$1109$1819$2113$3323$60]
```

```
SET $4E4=LShift
SET $4E6=RShift
SET $4E8=MultI
SET $4EA=DivI
SET $4EC=RemI
SET $4EE=SArgs
```

```
PROC ChkErr=*(BYTE r,b,eC)[$1610$88C0$8F0
$98$80C0$11F0
$4C Error$8A$4A4A$4A4A$98AA$9D EOF$60]
```

```
PROC Break1=*(BYTE err)
[$1A2$1186$48$20 Break$68$A8$60]
```

```
PROC Open=*(BYTE d,BYTE ARRAY f,BYTE m,a2)
[$48$A186$A284$A8$A9$0$99 EOF$A8$A1B1$8D OpenBuf $A8$C8$9BA9$2D0$A1B1$99 OpenBuf $88$F
$68$A2 OpenBufL $A0 OpenBufH $20Opn$4C ChkErr]
```

```
PROC PrintE=*(BYTE ARRAY s)[$A186$AA$A1A4$A5device]
PROC PrintDE=*(BYTE d,BYTE ARRAY s)
[$20 Prt$4C ChkErr]
```

```
PROC Close=*(BYTE d)[$20 Clos$4C ChkErr]
```

```
PROC Print=*(BYTE ARRAY s)[$A186$AA$A1A4$A5device]
PROC PrintD=*(BYTE d,BYTE ARRAY s)
[$20Outputq$4C ChkErr]
```

```
PROC InS=*()
[$20In$A084$BD$348$3F0$38$1E9$A0$0$A591$A0A4$60]
```

```
PROC InputS=*(BYTE ARRAY s)[$A286$AA$A2A4$A5device]
PROC InputSD=*(BYTE d,BYTE ARRAY s)[$48$FFA9$A385$68]
```

```
PROC InputMD=(BYTE d,BYTE ARRAY s,BYTE m)
[$48$A186$A284$A0$0$A3A5$A191$68$A2A4]
PROC InputD=(BYTE d,BYTE ARRAY s)[$20InS$4C ChkErr]
```

```
BYTE FUNC GetD=(BYTE d)[$7A2]
PROC CCIO=*( )
[$A486$A0A$A0A$AA$A4A5$9D$342$A9$0$9D$348$9D$349
$98$20$E456$A085$4C ChkErr]
```

```
PROC PutE=*( )[$A9$9B]
PROC Put=(BYTE c)[$AA$A5device]
PROC PutD=(BYTE d,BYTE c)[$A186$A1A4]
PROC PutD1=*( )[$BA2$4C CCIO]
```

```
PROC PutDE=(BYTE dev)[$A0$9B$F7D0]
```

```
PROC XIO=(BYTE d,f,c,a1,a2,BYTE ARRAY s)
[$20XIOstr$4C ChkErr]
```

```
PROC CToStr=*( )
[$D485$D586$20$D9AA$20$D8E6$FFA0$A2$0$C8$E8
$F3B1$9D$550$F710$8049$9D$550$8E$550$60]
```

```
PROC PrintB=(BYTE n)[$A2$0]
PROC PrintC=(CARD n)[$20 CToStr$A5device]
PROC PNum=*( )[$50A2$5A0$20 Outputq$4C ChkErr]
```

```
PROC PrintBE=(BYTE n)[$A2$0]
PROC PrintCE=(CARD n)[$20PrintC$4CPutE]
```

```
PROC PrintBD=(BYTE d, n)[$A0$0]
PROC PrintCD=(BYTE d, CARD n)
[$A085$8A$A284$A2A6$20 CToStr$A0A5$4CPNum]
```

```
PROC PrintBDE=(BYTE n)[$A0$0]
PROC PrintCDE=(BYTE d,CARD n)
[$20PrintCD$A0A5$4CPutDE]
```

```
BYTE FUNC InputB=*( )
CARD FUNC InputC=*( )
INT FUNC InputI=*( )
[$A5 device]
BYTE FUNC InputBD=(BYTE d)
CARD FUNC InputCD=(BYTE d)
INT FUNC InputID=(BYTE d)
[$13A2$8E$550$50A2$5A0$20InputD$50A9$5A2]
```

```
MODULE
```

```
BYTE CIO_status
```

```
CHAR FUNC CIOQ=(BYTE dev, CARD addr,
size, BYTE cmd, aux1, aux2)
[$29$F$85$A0$86$A1$A$A$A$A$A$A$A$A5$A5
$9D$342$A5$A3$9D$348$A5$A4$9D$349
$A5$A6$F0$8$9D$34A$A5$A7$9D$34B$98
$9D$345$A5$A1$9D$344$20$E456
$8C CIO_status$C0$88$D0$6$98$A4$A0
$99 EOF$A085$60]
```

```
CARD FUNC Bget=(BYTE dev,
                CARD addr, size)
[$48$A9$7$85$A5$A9$0$85$A6$A5$A3$5$A4
$D0$6$85$A0$85$A1$68$60$68$20 CIOQ
$BD$348$85$A0$BD$349$85$A1$60]
```

```
MODULE ; for user
```

```
;-----
```

```
CARD comtab=$0A
```

```
BYTE errnum
```

```
PROC Dosing ()
```

```
[ $6C $0C $00 ]
```

```
BYTE FUNC Find (BYTE ARRAY str2,str1)
```

```
BYTE len1,len2,z1,z2,flg,pos
```

```
IF str1(0)>=str2(0) THEN
```

```
len2=str2(0)
```

```
len1=str1(0)
```

```
len1== -len2+1
```

```
z1=0
```

```
z2=0
```

```
DO
```

```
flg=$FF
```

```
z1==+1
```

```
FOR z2=1 to len2
```

```
DO
```

```
IF str1(z1+z2-1)#str2(z2) THEN
```

```
flg=0
```

```
FI
```

```
OD
```

```
UNTIL z1=len1 OR flg#0
```

```
OD
```

```
IF flg#0 THEN
```

```
pos=z1
```

```
ELSE
```

```
pos=0
```

```
FI
```

```
ELSE
```

```
pos=0
```

```
FI
```

```
RETURN (pos)
```

```
BYTE FUNC Sparta ()
```

```
BYTE r = [1]
```

```
BYTE POINTER bp
```

```
bp = comtab
```

```
IF bp^ # $4C THEN
```

```
r = 0
```

```

FI

RETURN (r)

PROC ZCRNAME=$1356 () ; DUMMY

PROC GetCRNAME(BYTE ARRAY s)
  BYTE i

  i=1
  DO
    s(i)=Peek(comtab+32+i)
    i==+1
  UNTIL s(i-1)=$9B
  OD
  s(0)=i-2
RETURN

PROC GetCRPar(BYTE ARRAY s)
  BYTE i
  i=1
  DO
    s(i)=Peek(comtab+35+i)
    i==+1
  UNTIL s(i-1)=$9B
  OD
  s(0)=i-2
RETURN

PROC Err (BYTE num)
  errnum = num
RETURN

PROC Main()

  BYTE ARRAY fn(20)
  BYTE ARRAY st(40)
  BYTE ARRAY ds(120)
  BYTE ARRAY s(255)
  BYTE fnd = [0]
  BYTE di
  CARD si
  Error=Err

  IF Sparta()==0 THEN
    Print ("Filename: ")
    InputS (fn)
    PutE ()
    Print ("Searchstring: ")
    InputS (st)
  ELSE
    ZCRNAME=comtab+3
    ZCRNAME()
    GetCRNAME(fn)
    ZCRNAME()
    GetCRPar(st)
  FI

  Open (1,fn, 4, 0)

```

```
errnum = 0
```

```
DO
```

```
  BGet (1,s+1,200)
```

```
  s(0)=200
```

```
  si = 1
```

```
DO
```

```
  fnd = 0
```

```
  ds(di)=s(si)
```

```
  si==+1
```

```
  di==+1
```

```
IF s(si)=13% $s(si)=155$  THEN
```

```
  ds(0)=di-1
```

```
  di=1
```

```
  IF s(si)=13 THEN
```

```
    si==+1
```

```
  FI
```

```
  si==+1
```

```
  fnd = Find(st,ds)
```

```
  IF fnd THEN
```

```
    PrintE(ds)
```

```
  FI
```

```
  FI
```

```
UNTIL si > 200
```

```
OD
```

```
UNTIL errnum#0
```

```
OD
```

```
Close (1)
```

```
RETURN
```