

Rev - a reverse search command for Acme

(version 0.1.1)

Alexander Sychev (santucco@gmail.com)

1. Introduction. This is quick-n-dirty implementation of **Rev** command for **Acme**. It behaves like **Look** but in reverse order.

2. Implementation.

```
// This file is part of rev
// Author Alexander Sychev
//
// Copyright (c) 2014, 2020 Alexander Sychev. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * The name of author may not be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
package main
import(
    <Imports 4>
)
```

3.

```
func main(){
    <Obtaining of id of a window 5>
    <Opening the window by id 7>
    <Initial search 9>
    <Processing of events 17>
}
```

4.

```
<Imports 4> ≡
"os"
"strconv"
```

See also sections 6 and 8.

This code is used in section 2.

5.

```

⟨Obtaining of id of a window 5⟩ ≡
  id, err := strconv.Atoi(os.Getenv("winid"))
  if err ≠ nil {
    return
  }

```

This code is used in section 3.

6.

```

⟨Imports 4⟩ +≡
  "github.com/santucco/goacme"

```

7.

```

⟨Opening the window by id 7⟩ ≡
  w, err := goacme.Open(id)
  if err ≠ nil {
    return
  }

```

This code is used in section 3.

8.

```

⟨Imports 4⟩ +≡
  "strings"

```

9. If a string to be looked for is specified in command line, we should use it.

```

⟨Initial search 9⟩ ≡
  if len(os.Args) > 1 { s := strings.Join(os.Args[1:], " ")
    ⟨Make a reverse search of s 15⟩
  } else ⟨Look for selected string 10⟩
  ⟨Set dot 16⟩

```

This code is used in section 3.

10.

```

⟨Look for selected string 10⟩ ≡
  {
    ⟨Set the addr address to dot 11⟩
    ⟨Read the addr address into b, e 12⟩
    ⟨Read selected string from "xdata" file to s 13⟩
    ⟨Set the addr address to b, e 14⟩
    ⟨Make a reverse search of s 15⟩
  }

```

This code is used in sections 9 and 17.

11.

```

⟨Set the addr address to dot 11⟩ ≡
  if w.WriteCtl("addr=dot") ≠ nil {
    return
  }

```

This code is used in section 10.

12.

⟨Read the addr address into b, e 12⟩ \equiv
 $b, e, err := w.ReadAddr()$
if $err \neq \text{nil}$ {
 return
}

This code is used in section 10.

13.

⟨Read selected string from "xdata" file to s 13⟩ \equiv
 $s := ""$
{
 $d, err := w.File("xdata")$
 if $err \neq \text{nil}$ {
 return
 }
 $buf := \text{make}([]\text{byte}, e - b + 1)$
 for $n, _ := d.Read(buf); n > 0$; $n, _ = d.Read(buf)$ {
 $s += \text{string}(buf[:n])$
 }
}

This code is used in section 10.

14.

⟨Set the addr address to b, e 14⟩ \equiv
if $w.WriteAddr(\text{"\%d, \%d"}, b, e - \text{len}(s)) \neq \text{nil}$ {
 return
}

This code is used in sections 10 and 18.

15. Reverse search is processed by writing " $?<\text{regex}>?$ " to "addr" file, but before regex-specific symbols of s have to be escaped

⟨Make a reverse search of s 15⟩ \equiv
{
 $es := ""$
 for $_, v := \text{range } s$ {
 if $strings.ContainsRune("\\[/[] .+?()*\^$", v)$ {
 $es += "\\\"$
 }
 $es += \text{string}(v)$
 }
 if $w.WriteAddr(\text{"\%s?"}, es) \neq \text{nil}$ {
 return
 }
}

This code is used in sections 9, 10, and 18.

16.

```

⟨Set dot 16⟩ ≡
  if w.WriteCtl("dot=addr\nshow") ≠ nil {
    return
  }

```

This code is used in sections 9 and 17.

17. Events is being read from *w* and processed in next manner:

- for text manipulating events the program is finished
- for command executions "Rev" command is processed, "Look" command finishes the program, others are sent back to Acme.
- looking is processed in reverse order
ev can contain a needed string in *ev.Arg* for "Rev" command and in *ev.Text* and *ev.Arg* for looking event. In the latter case non-empty *ev.Arg* is joined to *s*

```

⟨Processing of events 17⟩ ≡
  { for ev, err := w.ReadEvent();
    err ≡ nil;
    ev, err = w.ReadEvent() { if ev.Type ≡ goacme.Insert ∨ ev.Type ≡ goacme.Delete {
      return
    }
    switch ev.Type & ^goacme.Tag { case goacme.Execute: switch ev.Text { case "Look":
      w.UnreadEvent(ev)
      return
    case "Rev": if len(ev.Arg))0 { b := ev.Begin
      e := ev.End
      s := ev.Arg
      ⟨Look for s at addr b, e 18⟩
    } else ⟨Look for selected string 10⟩
    }
    ⟨Set dot 16⟩
    continue
  }
  w.UnreadEvent(ev)
  case goacme.Look:
    b := ev.Begin
    e := ev.End
    s := ev.Text
    if len(ev.Arg))0 {
      s += "␣" + ev.Arg
    }
    ⟨Look for s at addr b, e 18⟩
    ⟨Set dot 16⟩
    continue
  }
}

```

This code is used in section 3.

18.

⟨Look for s at addr b, e 18⟩ ≡
 ⟨Set the addr address to b, e 14⟩
 ⟨Make a reverse search of s 15⟩

This code is used in section 17.

?iregex?: 15.
addr: 15.
Arg: 17.
Args: 9.
Atoi: 5.
Begin: 17.
buf: 13.
ContainsRune: 15.
Delete: 17.
End: 17.
err: 5, 7, 12, 13, 17.
es: 15.
ev: 17.
Execute: 17.
File: 13.
Getenv: 5.
goacme: 6, 7, 17.
id: 5, 7.
Insert: 17.
Join: 9.
Look: 17.
main: 2, 3.
Open: 7.
os: 4, 5, 9.
Read: 13.
ReadAddr: 12.
ReadEvent: 17.
Rev: 17.
strconv: 4, 5.
strings: 8, 9, 15.
Tag: 17.
Text: 17.
Type: 17.
UnreadEvent: 17.
WriteAddr: 14, 15.
WriteCtl: 11, 16.
xdata: 10, 13.

- ⟨Imports [4](#), [6](#), [8](#)⟩ Used in section [2](#).
- ⟨Initial search [9](#)⟩ Used in section [3](#).
- ⟨Look for selected string [10](#)⟩ Used in sections [9](#) and [17](#).
- ⟨Look for s at addr b, e [18](#)⟩ Used in section [17](#).
- ⟨Make a reverse search of s [15](#)⟩ Used in sections [9](#), [10](#), and [18](#).
- ⟨Obtaining of id of a window [5](#)⟩ Used in section [3](#).
- ⟨Opening the window by id [7](#)⟩ Used in section [3](#).
- ⟨Processing of events [17](#)⟩ Used in section [3](#).
- ⟨Read selected string from "**xdata**" file to s [13](#)⟩ Used in section [10](#).
- ⟨Read the addr address into b, e [12](#)⟩ Used in section [10](#).
- ⟨Set dot [16](#)⟩ Used in sections [9](#) and [17](#).
- ⟨Set the addr address to dot [11](#)⟩ Used in section [10](#).
- ⟨Set the addr address to b, e [14](#)⟩ Used in sections [10](#) and [18](#).

Rev - a reverse search command for Acme

(version 0.1.1)

	Section	Page
Introduction	1	2
Implementation	2	3

Copyright © 2014, 2020 Alexander Sychev. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name of author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.