



Ubiquitous remind Calendar

Serge Y. Stroobandt

Copyright 2014–2015, licensed under [Creative Commons BY-NC-SA](#)

This page is still under construction.

Introduction

- text-based
- Distributed revision control
- spy-safe, like those [personal information managers for contacts and tasks](#) that I use.
- semantic text, much more powerful. program dates into your calendar until the end of times. Once you know it a bit, quicker.

David F. Skoll started writing Remind in 1989 because I was fed up with the limitations of the UNIX calendar program.

Documentation

scattered

home page

[developer David F. Skoll of Roaring Penguin](#)

Remind tutorial

[Remind tutorial slides](#)

manual

[man page](#)

wikis

- [Further Information on the Wiki](#)

- [43 Folders](#)

FAQ

[43 Folders remind FAQ](#)

articles

- 2000 Linux Journal [Remind: The Ultimate Personal Calendar](#)
- 2007 really sparked interest among Mac OS X users [43 Folders Guest: Mike Harris looks at "Remind"](#)
- It is funny how I keep on running into [Aditya Mahajan](#) —well, after all we are both electronic engineers— [Finally, a usable calendar program](#)
- [Remind – A Killer App](#)
- [HOWTO: One remind to rule them all](#)
- [Manage your time with Remind](#)

Installation

[ssh-key](#)

```
user@client $ sudo apt-get update
user@client $ sudo apt-get install remind wyrd
user@client $ echo 'export EDITOR="nano"' >> .bashrc
user@client $ source .bashrc
```

Distributed revision control

```
user@server $ sudo apt-get update
user@server $ sudo apt-get install remind wyrd
user@server $ hg init remind
user@server $ touch ~/remind/reminders
user@server $ cp /etc/wyrdrc ~/remind/
user@server $ hg add
user@server $ hg com -m "first commit"
```

Symbolic links

```
user@client $ hg clone ssh://user@server.url:port/remind
user@client $ ln -s remind/reminders .reminders
user@client $ ln -s remind/wyrdrc .wyrdrc
user@client $ nano .wyrdrc
```

```
# the default reminder file to display
set reminders_file="$HOME/.remind/reminders"
```

Wyrd

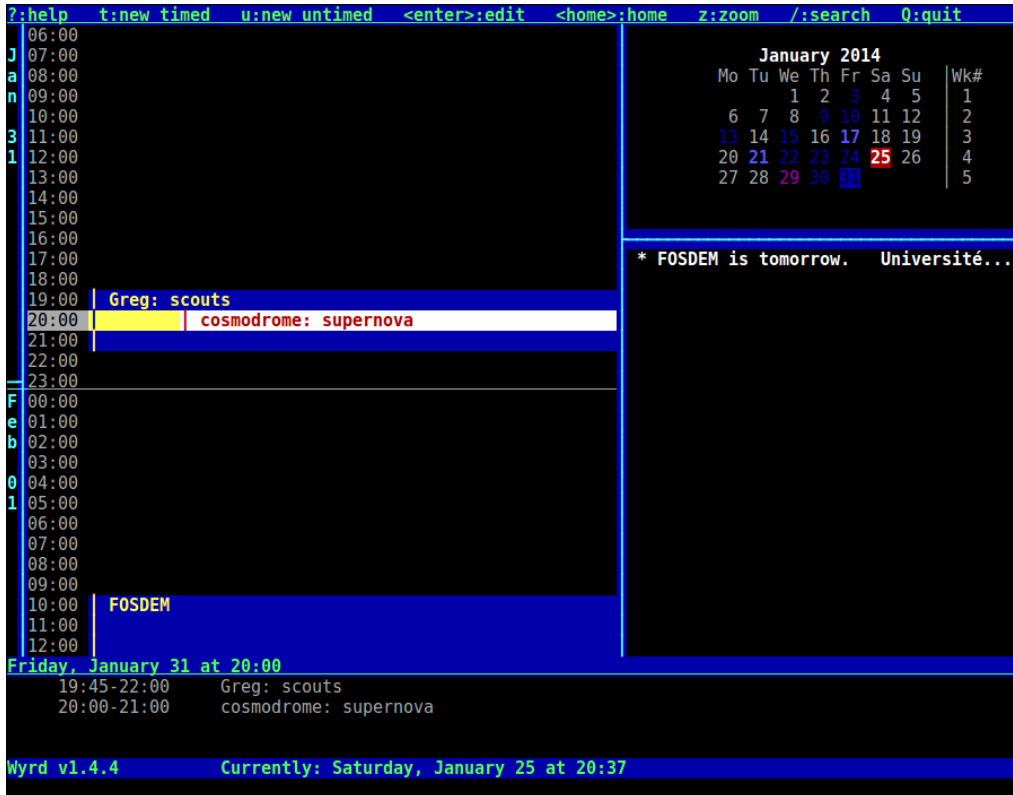


Figure 1: A screen-shot of wyrd

~/wyrdrdrc for Europeans

Most ~/wyrdrdrc examples found on the Internet are typical US examples. Hence, it will not hurt presenting a typical European .wyrdrdrc example here. Furthermore, I have added a number of key bindings rendering Wyrdr more accessible to non-vi(m)-users.

```

# Wyrd run-configuration file

# command for the Remind executable
set remind_command="remind"
# the default reminder file to display
set reminders_file="$HOME/remind/reminders"
# command for editing an old appointment, given a line number %line% and
filename %file%
set edit_old_command="${VISUAL:-$EDITOR} +%line% %file% > /dev/null 2>&1"
# command for editing a new appointment, given a filename %file%
set edit_new_command="${VISUAL:-$EDITOR} +999999 %file% > /dev/null 2>&1"
# command for free editing of the reminders file, given a filename %file%
set edit_any_command="${VISUAL:-$EDITOR} %file% > /dev/null 2>&1"

# templates for creating new appointments
# %monname% -> month name, %mon% -> month number, %mday% -> day of the
month,
# %year% -> year, %hour% -> hour, %min% -> minute, %weekdayname% -> weekday name
# %yday% -> weekday number
set timed_template="REM %monname% %mday% %year% AT %hour%:%min% DURATION
1:00 MSG %1, %3: %\"%\"%"
set untimed_template="REM %monname% %mday% %year% MSG "

# weekly recurrence
set template0="REM %weekdayname% AT %hour%:%min% DURATION 1:00 MSG "
set template1="REM %weekdayname% MSG "

# monthly recurrence
set template2="REM %mday% AT %hour%:%min% DURATION 1:00 MSG "
set template3="REM %mday% MSG "

# algorithm to use for determining busy level
# "1" -> count the number of reminders in each day
# "2" -> count the number of hours of reminders in each day
set busy_algorithm="2"
# for busy_algorithm="2", assume that untimed reminders occupy this many
minutes
set untimed_duration="0"

# if busy_algorithm="1", number of reminders per day allowed for each
calendar
# colorization level; if busy_algorithm="2", use number of hours of reminders
# per day
set busy_level1="0" # level1 color
set busy_level2="3" # level2 color
set busy_level3="5" # level2 color, bold
set busy_level4="7" # level3 color
# (everything else is level3 color, bold)

# first day of the week is Sunday
set week_starts_monday="true"

# 12/24 hour time settings
set schedule_12_hour="false"
set selection_12_hour="false"
set status_12_hour="false"
set description_12_hour="false"

# whether or not to keep the cursor centered when scrolling through timed
# reminders

```

~/bin/wyrd

```
user@client $ mkdir bin
user@client $ nano bin/wyrd
```

```
#!/bin/bash
VISUAL='/usr/bin/gvim -f --servername REMIND --remote-tab '

cd $HOME/remind
clear
echo

hg pull -u server || { echo; hg pull -u roam; }
echo

if [ -f /usr/bin/wmctrl ]; then
    wmctrl -R 'Terminal'
fi

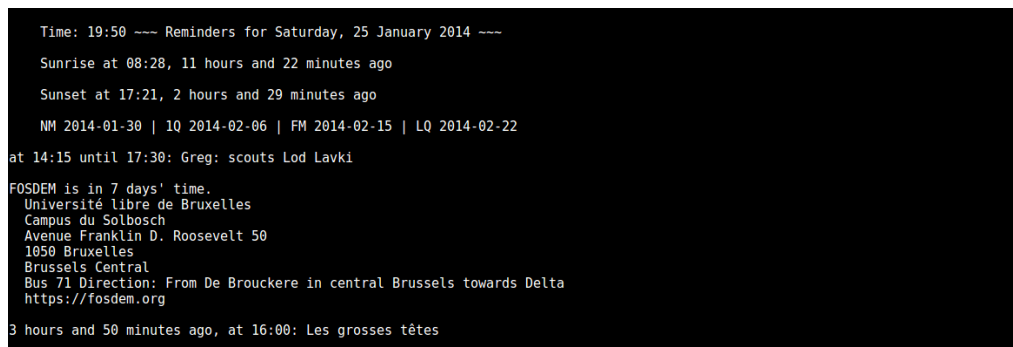
/usr/bin/wyrd && echo && hg addremove && hg commit -m '~/bin/wyrd' -u
$USER@$HOSTNAME && { hg push server || { echo; hg push roam; } }
# The last && avoids a push when nothing has changed.
echo

read -n1 -r -p 'Press any key to quit...'
clear

exit 0
```

```
user@client $ chmod +x bin/wyrd
user@client $ ./bin/wyrd
```

Examples



```
Time: 19:50 --- Reminders for Saturday, 25 January 2014 ---
Sunrise at 08:28, 11 hours and 22 minutes ago
Sunset at 17:21, 2 hours and 29 minutes ago
NM 2014-01-30 | 1Q 2014-02-06 | FM 2014-02-15 | LQ 2014-02-22
at 14:15 until 17:30: Greg: scouts Lod Lavki
FOSDEM is in 7 days' time.
Université libre de Bruxelles
Campus du Solbosch
Avenue Franklin D. Roosevelt 50
1050 Bruxelles
Brussels Central
Bus 71 Direction: From De Brouckere in central Brussels towards Delta
https://fosdem.org
3 hours and 50 minutes ago, at 16:00: Les grosses têtes
```

Figure 2: An example of the output of the `rem` command. The example includes among others reminders for ephemerids and the yearly **FOSDEM** conference in Brussels (see below).

Top-level reminders file

```
RUN OFF

BANNER %_      Time: %# ~~~ Reminders for %w, %d %m %y ~~~

include /home/serge/remind/ephemerids
include /home/serge/remind/greg
include /home/serge/remind/seasons
include /home/serge/remind/dst.eu
include /home/serge/remind/holidays.be-nl
include /home/serge/remind/holidays.us
include /home/serge/remind/vacations.be-nl

include /home/serge/remind/birthdays
include /home/serge/remind/admin
include /home/serge/remind/health
include /home/serge/remind/trash
include /home/serge/remind/clubs
include /home/serge/remind/floss
include /home/serge/remind/contests
include /home/serge/remind/streams

# include /home/serge/remind/2013
include /home/serge/remind/2014
include /home/serge/remind/2015
include /home/serge/remind/2016

REM Mon MSG Week [weekno()]

# REM 201 AT :00 DURATION 1:00 MSG %1, %3: %%"
```

FOSDEM

```
REM Sat 1 Feb +7 AT 10:30 DURATION 8:30 MSG %"FOSDEM%" is %b.\
%_Université libre de Bruxelles\
%_Campus du Solbosch\
%_Avenue Franklin D. Roosevelt 50\
%_1050 Bruxelles\
%_Brussels Central\
%_Bus 71 Direction: From De Brouckere in central Brussels towards Delta\
%_https://fosdem.org
REM Sun 1 Feb AT 9:00 DURATION 9:00 MSG FOSDEM
```

Ham Radio fair

Every year, on the last Friday of June starts the three day long *Ham Radio* fair in Friedrichshafen, south Germany. One would feel tempted to program this reminder as falling into the fourth week of June. However, this is not valid for every year. There might be a fifth Friday in June! To circumvent this problem, one applies the technique called *backward scanning* as described in the [Remind tutorial slides](#). This technique consist in scanning for the first Friday of the next month —July— and then counting back exactly seven days to find the last Friday of the previous month —June. The subsequent days are marked by counting back less days.

```
;Backward scanning for the last Friday of June
REM Fri Jul 1 --7 +30 AT 9:00 DURATION 9:00 MSG %"Ham Radio
Friedrichshafen%"
%_ http://www.hamradio-friedrichshafen.de/
REM Fri Jul 1 --6 AT 9:00 DURATION 9:00 MSG %"Ham Radio Friedrichshafen%"
REM Fri Jul 1 --5 AT 9:00 DURATION 6:00 MSG %"Ham Radio Friedrichshafen%"
```

Trash

```
REM Tue AT 19:45 BEFORE MSG %1, %3: PMD-, groen- en restafval%"%"
REM Wed AT 17:00 AFTER MSG %1, %3: Papier & karton%"%"
```

Ephemerids

```
;Hasselt

;longitude
SET $LongDeg -5
SET $LongMin 20
SET $LongSec 36

;latitude
SET $LatDeg 50
SET $LatMin 55
SET $LatSec 51

AT [sunrise(today())] MSG Sunrise at [sunrise(today())], %1%"%"

AT [sunset(today())] MSG Sunset at [sunset(today())], %1%"%"

MSG NM [moondate(0)] | 1Q [moondate(1)] | FM [moondate(2)] | LQ
[moondate(3)]%"%"
MSG ~~~~~%"%"
```


DST in Europe

```
SET Jan 1
SET Feb 2
SET Mar 3
SET Apr 4
SET May 5
SET Jun 6
SET Jul 7
SET Aug 8
SET Sep 9
SET Oct 10
SET Nov 11
SET Dec 12
FSET last(mo) "1 " + mon((mo%12)+1)+" --7"
```

```
;http://nl.wikipedia.org/wiki/Zomertijd#Europa
```

```
# de zomertijd begint op de laatste zondag van maart, als de klok om 01:00
UTC (Nederlandse/Belgische tijd: 02:00 uur) een uur vooruit wordt gezet,...
REM Sun [last(Mar)] AT 02:00 MSG %"Begin zomertijd%": 02u wordt 03u lokale
tijd.
```

```
# ...en eindigt op de laatste zondag van oktober, als de klok om 01:00 UTC
(Nederland/België: 03:00 uur zomertijd) een uur terug wordt gezet.
REM Sun [last(Oct)] AT 03:00 MSG %"Einde zomertijd%": 03u wordt 02u lokale
tijd.
```

Belgian holidays

```

SET yr year(today())
SET easter easterdate(yr)

;http://nl.wikipedia.org/wiki/Feestdagen_in_België

;Wettelijke feestdagen
OMIT Jan 01
REM Jan 01 MSG Nieuwjaar

OMIT [TRIGGER(easter)]
REM [TRIGGER(easter)] MSG Pasen

OMIT [TRIGGER(easter+1)]
REM [TRIGGER(easter+1)] MSG Paasmaandag

OMIT May 01
REM May 01 MSG Dag van de Arbeid

OMIT [TRIGGER(easter+39)]
REM [TRIGGER(easter+39)] MSG Onze-Lieve-Heer-Hemelvaart

OMIT [TRIGGER(easter+49)]
REM [TRIGGER(easter+49)] MSG Pinksteren

OMIT [TRIGGER(easter+50)]
REM [TRIGGER(easter+50)] MSG Pinkstermaandag

OMIT Jul 21
REM Jul 21 MSG Nationale feestdag van België

OMIT Aug 15
REM Aug 15 MSG Onze-Lieve-Vrouw-Hemelvaart

OMIT Nov 01
REM Nov 01 MSG Allerheiligen

OMIT Nov 11
REM Nov 11 MSG Wapenstilstand

OMIT Dec 25
REM Dec 25 MSG Kerstmis

;Ambtenaren
#OMIT Nov 02
REM Nov 02 MSG Allerzielen%"%"

#OMIT Nov 15
REM Nov 15 MSG Koningsdag%"%"

#OMIT Dec 26
REM Dec 26 MSG Tweede Kerstdag%"%"

;Gemeenschapsfeestdag
#OMIT
REM Jul 11 MSG Feestdag van Vlaanderen%"%"
REM Sep 27 MSG Dag van de Franse Gemeenschap%"%"
REM Nov 15 MSG Dag van de Duitstalige Gemeenschap%"%"

;Profane vieringen
REM Feb 14 +14 MSG %"Valentijnsdag%" 15 %b.
REM Apr 01 MSG 1 April%"%"

```

Black Friday & Cyber Monday

SET week1 01
SET week2 08
SET week3 15
SET week4 22

REM Nov 11 MSG Aliexpress sale

REM Thu [week4] Nov MSG Thanksgiving

REM Fri [week4+1] Nov MSG Black Friday!

REM Mon [Week4+4] Nov MSG Cyber Monday!

all US holidays (including Jewish)

Flemish school vacations

```

SET yr year(today())
SET easter easterdate(yr)

;http://www.ond.vlaanderen.be/infolijn/faq/schoolvakanties/

;De zomervakantie begint op 1 juli en eindigt op 31 augustus.
REM [date(yr,07,01)] *1 UNTIL [date(yr,08,31)] MSG zomervakantie

;De herfstvakantie begint op maandag van de week waarin 1 November valt en
duurt 1 week.
;Als 1 November op zondag valt, dan begint de herfstvakantie op 2 November.
REM Mon 3 Nov --9 MSG herfstvakantie
REM Mon 3 Nov --8 MSG herfstvakantie
REM Mon 3 Nov --7 MSG herfstvakantie
REM Mon 3 Nov --6 MSG herfstvakantie
REM Mon 3 Nov --5 MSG herfstvakantie
REM Mon 3 Nov --4 MSG herfstvakantie
REM Mon 3 Nov --3 MSG herfstvakantie
REM Mon 3 Nov --2 MSG herfstvakantie
REM Mon 3 Nov --1 MSG herfstvakantie

;De kerstvakantie begint op maandag van de week waarin kerstdag valt en
duurt 2 weken.
;Als kerstdag op zaterdag of zondag valt, dan begint de kerstvakantie op
maandag na kerstdag.
REM Mon 4 Jan --16 MSG kerstvakantie
REM Mon 4 Jan --15 MSG kerstvakantie
REM Mon 4 Jan --14 MSG kerstvakantie
REM Mon 4 Jan --13 MSG kerstvakantie
REM Mon 4 Jan --12 MSG kerstvakantie
REM Mon 4 Jan --11 MSG kerstvakantie
REM Mon 4 Jan --10 MSG kerstvakantie
REM Mon 4 Jan --9 MSG kerstvakantie
REM Mon 4 Jan --8 MSG kerstvakantie
REM Mon 4 Jan --7 MSG kerstvakantie
REM Mon 4 Jan --6 MSG kerstvakantie
REM Mon 4 Jan --5 MSG kerstvakantie
REM Mon 4 Jan --4 MSG kerstvakantie
REM Mon 4 Jan --3 MSG kerstvakantie
REM Mon 4 Jan --2 MSG kerstvakantie
REM Mon 4 Jan --1 MSG kerstvakantie

;De krokusvakantie begint de 7de maandag voor Pasen en duurt 1 week.
REM [trigger(easter-50)] *1 UNTIL [trigger(easter-42)] MSG krokusvakantie

;De paasvakantie begint de 1ste maandag van april en duurt 2 weken.
REM Mon 14 Apr --16 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --15 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --14 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --13 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --12 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --11 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --10 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --9 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie
REM Mon 14 Apr --8 SATISFY [easter > date(yr,3,31) && easter <=
date(yr,4,15)] MSG paasvakantie

```

Additional scripts

```
$ nano ~/.bashrc  
  
alias rem='clear; rem'
```

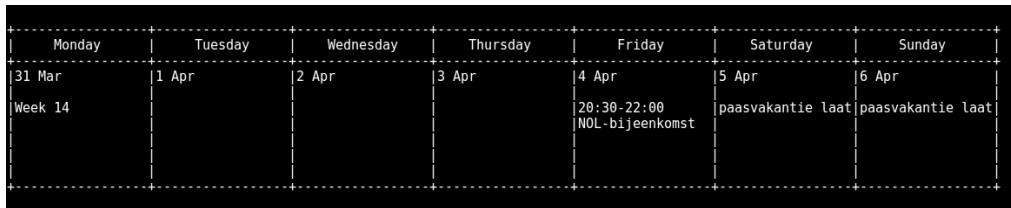
The `rem+` prints reminders for coming days:

```
$ nano bin/rem+  
  
#!/bin/bash  
clear  
rem $(date --date="$1 day" +%Y-%m-%d)
```

command	reminders for
<code>rem+</code>	tomorrow
<code>rem+ 2</code>	day after tomorrow
<code>rem+ n</code>	n th day after today

The `wk` script produces a calendar for this week or more weeks:

```
$ nano bin/wk  
  
#!/bin/bash  
clear  
rem -c+$1 -b1 -m -w$(tput cols) $2 $3
```



Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
31 Mar	1 Apr	2 Apr	3 Apr	4 Apr	5 Apr	6 Apr
Week 14				20:30-22:00 NOL-bijeenkomst	paasvakantie laat	paasvakantie laat

Figure 3: Example output of the `wk` script

The `wk+` script produces calendars for coming weeks:

```
$ nano bin/wk+  
  
#!/bin/bash  
clear  
rem -c+ -b1 -m -w$(tput cols) $(date --date="$1 week" +%F)
```

command	calendar for
wk+	next week
wk+ 2	second week after this week
wk+ n	n th week after this week

The `mo` script produces a six-week calendar, starting with the current week. This view will always encompass an entire calendar month.

```
$ nano bin/mo

#!/bin/bash
clear
rem -c+6 -b1 -m -w$(tput cols) $1 $2
```

The `mo+` script

```
$ nano bin/mo

#!/bin/bash
clear
if [ $# -eq 0 ]
then
    MONTHS=1
else
    MONTHS=$1
fi
mo $(date --date="$MONTHS months" +%F)
```

Hipster PDA

From [Merlin Mann](#) the same [Folders 43](#), [Hipster PDA](#)

I chose [ConTeXt](#) to render the PDF because the result ends up nicer than with [LaTeX](#). The *Symbola* font allows for a wide variety of calendar symbols. It is available through the `ttf-ancient-fonts` package. Be sure to also follow the [LuaTeX font installation instructions](#), as well as the [instructions to install simplefonts](#).

```
$ nano bin/hpda
```



```
#!/bin/bash
# Creates a hipster PDA.
# https://en.wikipedia.org/wiki/Hipster_PDA

echo
hg pull -u -R $HOME/remind
echo

rem -c+6 -b1 -m -w140 $1 $2 \
| pandoc --output=$HOME/hpda.tex -t context --standalone \
-V mainlang='nl' -V papersize='A4, landscape' \
-H <(echo '\setuppagenumbering[location=]'; \
echo '\setupbodyfontenvironment[default][em=italic]'; \
echo '\usemodule[simplefonts][size=10pt]'; \
echo '\setmainfont[Symbola]'; \
)

context $HOME/hpda.tex --result=~/.hpda.pdf --batchmode > /dev/null 2>&1

rm $HOME/hpda.{tex,tuc,log}

xdg-open $HOME/hpda.pdf > /dev/null 2>&1
```

Notifications with cron

I have not been using notifications so far.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Other licensing available on request.

Unattended [CSS](https://www.princeps.com/) typesetting with .

This work is published at <https://hamwaves.com/remind/en/>.

Last update: Monday, March 1, 2021.