

Reserves automàtiques per l'estona de manteniment (des de 2022-08)

Els scripts i les tasques periòdiques programades (cron, crontab) que creaven les reserves setmanals varen estar en funcionament des de finals de 2013 fins principis de març de 2022. Llavors vàrem canviar la freqüència de les recàrregues dels imants i els scripts es varen haver d'adaptar.

La informació sobre els antics scripts i tasques periòdiques programades es troba a la pàgina [Reserves automàtiques per l'estona de manteniment \(de 2013-10 a 2022-03\)](#).

IMPORTANT: Les reserves per les estones de manteniment es crearan automàticament amb vuit setmanes d'antelació, per tal que, si s'escau, hi hagi prou temps per modificar-les.

Instal·lar el paquet "dateutils"

Instal·lo el paquet dateutils per facilitar el càlcul amb dates:

```
$ sudo apt install dateutils
[sudo] password for sermnadmin: *****
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  dateutils
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 217 kB of archives.
After this operation, 1,555 kB of additional disk space will be used.
Get:1 http://ftp.es.debian.org/debian buster/main amd64 dateutils amd64
0.4.3-1 [217 kB]
Fetched 217 kB in 0s (586 kB/s)
Selecting previously unselected package dateutils.
(Reading database ... 253630 files and directories currently installed.)
Preparing to unpack .../dateutils_0.4.3-1_amd64.deb ...
Unpacking dateutils (0.4.3-1) ...
Setting up dateutils (0.4.3-1) ...
Processing triggers for install-info (6.5.0.dfsg.1-4+b1) ...
Processing triggers for man-db (2.8.5-2) ...
```

maintenance_bookings_2022.sh



cal adaptar el script.

Aquest es el programet que crea les reserves per l'estona de manteniment. S'executa tots els diumenges a les 3 de la matinada, i crea les reserves amb dues setmanes d'antelació per tal que es puguin modificar si s'escau. El programet també envia un correu electrònic indicant si ha pogut fer les reserves.

[maintenance_bookings_2022.sh](#)

```
#!/bin/bash
#
# maintenance_bookings_2022.sh
#
# (c) 2013, Miquel Cabanas, SeRMN, UAB
#
# Started writing: 2013/10/28
# Last updated: 2013/10/28
#
# This script runs the MySQL commands needed to automatically create
the
# bookings for the spectrometers maintenance tasks, mainly liquid
nitrogen
# refills.
#
# To run this script every Sunday 2am append an entry like,
#
# FIXME 0 3 * * Sun /opt/bmbee-114-1/bin/maintenance_nmr_bookings.sh
#
# to crontab. This entry means, at minute 0, hour 2, on every (*) day
of
# the month and on every (*) month of the year, if that day happens to
be
# a Sunday (day 0), execute the commands in the script file
# 'maintenance_nmr_bookings.sh' (this file).

# VARIABLES -----
----

# programs
#
date=`which date`
mail=`which mail`
mysql=`which mysql`

# email recipients and subject

to_address="miquel.cabanas@uab.cat"
#
cc_address="miriam.perez@uab.cat,pau.nolis@uab.cat,eva.monteagudo@uab.c
at"
cc_address="miquel.cabanas@gmail.com,miquel.cabanas@uab.es"
```

```

subject="SeRMN Booking System - Weekly maintenance reservations"

# date related variables. This script runs on Sunday, and bookings are
made
# eight weeks in advance (56 days) on Wednesday from 09:00 to 13:00.
Hence,
# we need to find those dates so that we can use them later when
booking time-slots.
#
# NOTE: these date variables are not currently used for booking
commands,
# only for the reporting email messages.

today_date=`$date --date='Today' +%Y/%m/%d` # 2013/10/28
mon_date_mail=`$date --date='8 week 3 day' +%a %Y/%m/%d` # Thu
2013/12/06
mon_weekno=`$date --date='8 week 3 day' +%V/%Y` # 52/2013

# Bumblebee booking system database where everything gets stored

myuser='*****'
mypass='*****'
mydb='*****'

# INSERT BOOKING ENTRIES IN THE DATABASE -----
-----

$mysql --database=$mydb --user=$myuser --password=$mypass << EOF
/* begin transaction */
BEGIN;
/* Insert bookings for weekly maintenance tasks in all vertical
magnets. */
/* By default, book Wednesday morning from 09:00 to 13:00, 8 weeks in
advance */
/* (i.e. 59 days) */

/* Avance DPX-250-QNP (250-AUT0) - Wednesday from 09:00 to 13:00 -> 4
hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 12, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* Avance DPX-250-BACS (250-ROBOT) - Wednesday from 09:00 to 13:00 -> 4
hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 15, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',

```

```
'Weekly maintenance tasks', 0);

/* Avance DPX-360 - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 1, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* AvanceIII 400SB - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 6, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* AvanceII 400WB - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 9, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* Avance DRX-500 - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 2, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* AvanceII+ 600 - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 7, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* Hypersense DNP - Wednesday from 09:00 to 13:00 -> 4 hours */
INSERT INTO bookings(bookwhen, duration, instrument, bookedby, userid,
projectid, discount, ip, comments, log, deleted) VALUES (date_add(
date_add( current_date(), interval 59 day), interval 9 hour),
'04:00:00', 17, 7, 7, 1, 0, '192.168.2.2', 'Weekly maintenance tasks',
'Weekly maintenance tasks', 0);

/* commit transaction */
COMMIT;
EOF
```

MAIL THE SYSTEM AND INSTRUMENT ADMINISTRATORS - -----

```
--  
#  
# Report to the system and instrument administrators whether Routine  
NMR  
# bookings have been created.  
#  
status=$?  
  
if [ $status -ne 0 ]; then  
    # mysql command failed. Send an email reporting the failure  
    #  
    $mail -s "$subject" -c $cc_address $to_address << EOM  
  
    AUTOMATED EMAIL MESSAGE. DO NOT REPLY  
  
    SeRMN - Universitat Autònoma de Barcelona  
  
    Failed to allocate time slots for weekly maintenance tasks  
    week $mon_weekno starting: $mon_date_mail.  
  
    Please, report this issue to the system administrator.  
  
EOM  
  
else  
    # mysql command succeeded. Send an email reporting that routine  
    # bookings have been created  
    #  
    $mail -s "$subject" -c $cc_address $to_address << EOM  
  
    AUTOMATED EMAIL MESSAGE. DO NOT REPLY  
  
    SeRMN - Universitat Autònoma de Barcelona  
  
    Time slots have been allocated for weekly maintenance tasks  
    on all SeRMN spectrometers for week $mon_weekno starting:  
    $mon_date_mail.  
  
EOM  
  
fi  
  
exit
```

El programet envia el següent correu electrònic si ha aconseguit fer les reserves,

```
----- Original Message -----  
Subject: SeRMN Booking System - Weekly maintenance reservations  
Date: Sun, 27 Oct 2013 16:54:46 +0300  
From: root <root@sermn.uab.cat>
```

To: Miquel.Cabanas@uab.cat

AUTOMATED EMAIL MESSAGE. DO NOT REPLY

SeRMN - Universitat Autònoma de Barcelona

Time slots have been allocated for weekly maintenance tasks on all SeRMN spectrometers for week 52/2013 starting: Mon 2013/12/23.

i aquest altre missatge en cas contrari,

----- Original Message -----

Subject: SeRMN Booking System - NMR Routine reservations

Date: Sun, 27 Oct 2013 16:54:46 +0300

From: root <root@sermn.uab.cat>

To: Miquel.Cabanas@uab.cat

AUTOMATED EMAIL MESSAGE. DO NOT REPLY

SeRMN - Universitat Autònoma de Barcelona

Failed to allocate time slots for weekly maintenance tasks week 52/2013 starting: Mon 2013/12/23.

Please, report this issue to the system administrator.

Entrada al registre crontab



cal adaptar la programació del script.

Aquesta és l'entrada creada al programa crontab per executar automàticament el programet que crear les reserves (crontab -u root -l):

```
# Bumblebee 1.1.4 - Automated bookings for weekly maintenance tasks. It runs
# on Sunday at 03:00.
#
0 3 * * Sun /opt/bmbee-114-1/bin/maintenance_bookings.sh
```

Seguretat



cal comprovar els permisos.

Per motius de seguretat (el programet conté el nom d'usuari i clau de pas de la base de dades de reserves) la carpeta `/opt/bmbee-114-1/bin/` està protegida i té l'accés restringit,

- permisos carpeta `bin`: 700
- permisos fitxer `maintenance_bookings.sh`: 700
- propietari (user:group): root:root

A més, per impedir el seu accés a través del servidor web, la carpeta `bin` conté un fitxer `.htaccess` amb el següent contingut:

```
order deny,allow
deny from all
```

Documentació

BASH date math

- [Doing Date Math on the Command Line, Part I | Linux Journal](#).
 - [Doing Date Math on the Command Line, Part II | Linux Journal](#).
- [Addition and subtraction arithmetics with Linux date command - Linux Tutorials - Learn Linux Configuration](#).
- [Math Arithmetic: How To Do Calculation in Bash?](#)
- [bash date arithmetic - Cerca de Google](#).

BASH modulus operator

- [How to Use Modulus in Bash?](#)
- [Operators](#). In *Advanced Bash-Scripting Guide. An in-depth exploration of the art of shell scripting*. Chapter 8. Operations and Related Topics, 8.1. Operators, by Mendel Cooper, 2014.

Dateutils

- [Dateutils](#) are a bunch of tools that revolve around fiddling with dates and times in the command line with a strong focus on use cases that arise when dealing with large amounts of financial data.
- [GitHub - hroptatyr/dateutils](#): nifty command line date and time utilities; fast date calculations and conversion in the shell.
- [Debian -- Package Search Results -- dateutils](#).

Cron

- [cron - Wikipedia](#).
- [Crontab.guru - The cron schedule expression editor](#).
- [Getting crontab events that would happen given a start date/time - Unix & Linux Stack Exchange](#).

From:
<https://sermn.uab.cat/wiki/> - **SeRMN Wiki**

Permanent link:
https://sermn.uab.cat/wiki/doku.php?id=sermn_wiki:userpages:marta:reserves_automatiques_manteniment_2022&rev=1661175070

Last update: **2022/08/22 15:31**

