rtcwake

The command that you are interested in is `rtcwake`:

> This program is used to enter a system sleep state until specified wakeup time.

testing

To find the correct syntax that works for you try the following:

```
sudo rtcwake -u -s 60 -m mem
```

This should suspend the computer for 60 seconds before restoring. The significant parameter is `mem` You have several options you can choose - play to find the value that works best for you:

```
          standby
                 ACPI state S1. This state offers  minimal,  though  real,
                 power savings, while providing a very low-latency transi-
                 tion back to a working system. This is the default mode.

          mem    ACPI state S3 (Suspend-to-RAM). This state offers signif-
                 icant  power  savings  as everything in the system is put
                 into a low-power  state,  except  for  memory,  which  is
                 placed in self-refresh mode to retain its contents.

          disk   ACPI  state  S4  (Suspend-to-disk). This state offers the
                 greatest power savings, and  can  be  used  even  in  the
                 absence  of  low-level platform support for power manage-
                 ment. This state operates  similarly  to  Suspend-to-RAM,
                 but  includes  a final step of writing memory contents to
                 disk.

          off    ACPI  state  S5  (Poweroff).  This  is  done  by  calling
                 '/sbin/shutdown'.   Not officially supported by ACPI, but
                 usually working.

          no     Don't suspend. The rtcwake command sets RTC  wakeup  time
                 only.

          on     Don't  suspend,  but  read  RTC  device  until alarm time
                 appears. This mode is useful for debugging.
```

suspend until a known time

A script (at the bottom of this post) could be used to suspend your computer and wake at a specific time:

syntax is `suspend_until [hh:mm]` for example

```
sudo ./suspend_until 07:30
```

Save the script as the name `suspend_until` and give it execute rights i.e.

```
chmod +x suspend_until
```

Cron

You can create a root cron job that calls this script to execute at a specific time in the evening and then awake in the morning:

```
sudo crontab -e
```

Now enter something like to run the suspend script at 23:30:

```
30 23 * * * /home/myhomefolder/suspend_until 07:30
```

suspend_until script

```
#!/bin/bash

# Auto suspend and wake-up script
#
# Puts the computer on standby and automatically wakes it up at specified time
#
# Written by Romke van der Meulen <redge.online@gmail.com>
# Minor mods fossfreedom for AskUbuntu
#
# Takes a 24hour time HH:MM as its argument
# Example:
# suspend_until 9:30
# suspend_until 18:45

# ------------------------------------------------------
# Argument check
if [ $# -lt 1 ]; then
    echo "Usage: suspend_until HH:MM"
    exit
fi

# Check whether specified time today or tomorrow
DESIRED=$((`date +%s -d "$1"`))
NOW=$((`date +%s`))
if [ $DESIRED -lt $NOW ]; then
    DESIRED=$((`date +%s -d "$1"` + 24*60*60))
fi

# Kill rtcwake if already running
sudo killall rtcwake
```

```
# Set RTC wakeup time
# N.B. change "mem" for the suspend option
# find this by "man rtcwake"
sudo rtcwake -l -m mem -t $DESIRED &

# feedback
echo "Suspending..."

# give rtcwake some time to make its stuff
sleep 2

# then suspend
# N.B. dont usually require this bit
#sudo pm-suspend

# Any commands you want to launch after wakeup can be placed here
# Remember: sudo may have expired by now

# Wake up with monitor enabled N.B. change "on" for "off" if
# you want the monitor to be disabled on wake
xset dpms force on

# and a fresh console
clear
echo "Good morning!"
```

N.B.

Change `mem` in this part of the script for whatever suspend method works for you:

```
# Set RTC wakeup time
sudo rtcwake -l -m mem -t $DESIRED &
```

You may also have to substitute the `-u` flag in place of the `-l` flag depending on whether your hardware clock uses UTC (`-u`) or local (`-l`) time. Note that your hardware clock is different from the system clock you see in your operating system.

credit to redgeonline