```
#! /usr/bin/bash
# script filename is mon.sh
#
# Wait until a specified user logs on
#
# Set up default values
mailopt=FALSE
interval=60
#
# process command options
while getopts mt: option
do
        case "$option"
        in
        m)      mailopt=TRUE;;
        t)      interval=$OPTARG;;
        \?)     echo "Usage: mon [-m] [-t n] user"
                echo " -m means to be informed by mail"
                echo " -t means check every n secs."
                exit 1;;
        esac
done
#
# Make sure a user name was specified
if [ "$OPTIND" -gt "$#" ]
then
        echo "Missing user name!"
        exit 2
fi
shiftcount=$((OPTIND – 1))
shift $shiftcount
user=$1
#
# Check for user logging on
until who | grep "^$user " > /dev/null
do
        sleep $interval
done
#
# When we reach this point, the user has logged on
if [ "$mailopt" = FALSE]
then
        echo "$user has logged on"
else
        runner=$(who am i | cut -c1-8)echo "$user has logged on" | mail $runner
fi
```

```
$ mon -m
```

```
$ mon -x fred
$ mon -m -t 600 ann &
Check every 10 min. for ann
[1] 5792
```

```
$ ls *.sh
$ ls *.sh > list
$ for i in `cat list`; do cp "$i" "$i".bak ; done
$ ls *.sh*
```

```
# This program lists the files in /sbin that are just plain text files, and possibly scripts:
$ for i in `ls /sbin`; do file /sbin/$i | grep ASCII; done
```

```
#!/bin/bash
# This script opens 4 terminal windows.
i="0"
while [ $i -lt 4 ]
do
        xterm &
        i=$[$i+1]
done
```

```
#!/bin/bash
# This generates a file every 5 minutes
while true; do
        touch pic-`date +%s`.jpg
        sleep 300
done
```

```
#!/bin/bash
# Calculate the average of a series of numbers.
SCORE="0"
AVERAGE="0"
SUM="0"
NUM="0"
while true; do
        echo -n "Enter your score [0-100%] ('q' for quit): "; read SCORE;
        if (("$SCORE" < "0")) || (("$SCORE" > "100")); then
                echo "Be serious. Common, try again: "
        elif [ "$SCORE" == "q" ]; then
                echo "Average rating: $AVERAGE%."
                break
        else
                SUM=$[$SUM + $SCORE]
                NUM=$[$NUM + 1]
                AVERAGE=$[$SUM / $NUM]
        fi
done
```

```
echo "Exiting."
```

```
#!/bin/bash
# This script copies files from my homedirectory into the webserver directory.
# A new directory is created every hour.
# If the pics are taking up too much space, the oldest are removed.
while true; do
	DISKFUL=$(df -h $WEBDIR | grep -v File | awk '{print $5 }' | cut -d "%" -f1 -)
	until [ $DISKFUL -ge "90" ]; do
		DATE=`date +%Y%m%d`
		HOUR=`date +%H`
		mkdir $WEBDIR/"$DATE"
		while [ $HOUR -ne "00" ]; do
			DESTDIR=$WEBDIR/"$DATE"/"$HOUR"
			mkdir "$DESTDIR"
			mv $PICDIR/*.jpg "$DESTDIR"/
			sleep 3600
			HOUR=`date +%H`
		done
		DISKFULL=$(df -h $WEBDIR | grep -v File | awk '{ print $5 }' | cut -d "%" -f1 -)
	done
	TOREMOVE=$(find $WEBDIR -type d -a -mtime +30)
	for i in $TOREMOVE; do
		rm -rf "$i";
	done
done
```

```
#!/bin/bash
# script filename is archiveoldstuff.sh
# This script creates a subdirectory in the current directory, to which old
# files are moved.
# Might be something for cron (if slightly adapted) to execute weekly or
# monthly.
ARCHIVENR=`date +%Y%m%d`
DESTDIR="$PWD/archive-$ARCHIVENR"
mkdir "$DESTDIR"
# using quotes to catch file names containing spaces, using read -d for more
# fool-proof usage:
find "$PWD" -type f -a -mtime +5 | while read -d $'\000' file
do
	gzip "$file"; mv "$file".gz "$DESTDIR"
	echo "$file archived"
done
```

```
#!/bin/bash
# This script provides wisdom
# You can now exit in a decent way.
FORTUNE=/usr/games/fortune
```

```
while true; do
	echo "On which topic do you want advice?"
	echo "1. politics"
	echo "2. startrek"
	echo "3. kernelnewbies"
	echo "4. sports"
	echo "5. bofh-excuses"
	echo "6. magic"
	echo "7. love"
	echo "8. literature"
	echo "9. drugs"
	echo "10. education"
	echo
	echo -n "Enter your choice, or 0 for exit: "
	read choice
	echo
	case $choice in
		1)
		$FORTUNE politics
		;;
		2)
		$FORTUNE startrek
		;;
		3)
		$FORTUNE kernelnewbies
		;;
		4)
		echo "Sports are a waste of time, energy and money."
		echo "Go back to your keyboard."
		echo -e "\t\t\t\t -- \"Unhealthy is my middle name\" Soggie."
		;;
		5)
		$FORTUNE bofh-excuses
		;;
		6)
		$FORTUNE magic
		;;
		7)
		$FORTUNE love
		;;
		8)
		$FORTUNE literature
		;;
		9)
		$FORTUNE drugs
		;;
		10)
		$FORTUNE education
		;;
```

```
                0)
                echo "OK, see you!"
                break
                ;;
                *)
                echo "That is not a valid choice, try a number from 0 to 10."
                ;;
        esac
done
```

```
#!/bin/bash
# filename is tolower.sh
# This script converts all file names containing upper case characters into
# file names containing lowercase characters
LIST="$(ls)"
for name in "$LIST"; do
        if [[ "$name" != *[[:upper:]]* ]]; then
                continue
        fi
        ORIG="$name"
        NEW=`echo $name | tr 'A-Z' 'a-z'`
        mv "$ORIG" "$NEW"
        echo "new name for $ORIG is $NEW"
done
----------------------------
# This program overwrites all the files it renames. So, try in a new directory.
$ touch test Test TEST
$ bash -x tolower.sh
```
