Each version of unix handles timezones a little differently. Ignoring timezones more or less, this comes fairly close

Code:
```
#! /usr/bin/ksh


#
#  convert unix time to a string
#
#   time="$(unixsecond2timestring $seconds)" is
#      similiar to the c construct:
#             strcpy(time,ctime(&xdate));
#      except that it ignores timezone considerations.
#      This means that it is exactly like:
#         strcpy(time,asctime(gmtime(&xdate)));
#
#      The only way to handle timezones is to figure out
#      your local number of seconds difference from GMT and
#      adjust the value of seconds before passing it.  This
#      means that for small values of "seconds" you may adjust
#      it to a negative number.  That's ok, this routine can
#      handle numbers in the range -86400 to 2147483647.
#
unixsecond2timestring() {
        integer uxsec mjd daysecond hour hoursecond minute second
        typeset -Z2 val
        typeset -R2 val2
        typeset -L3 fdow
        typeset dow time year month day
        typeset months
        set -A months xxx Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
        uxsec=$1

#
#  Calculate
#    mjd=modified julian day number  (range is 40586 - 65442)
#      Dec 31, 1969 has  mjd=40586
#      Jan 19, 2038 has  mjd=65442
#    daysecond=number of second during the day (range is 0 - 86399)
#    hoursecond=number of second during hour   (range is 0 - 3599)
#    hour, minute, second represent current time
#
        ((mjd=(uxsec/86400)+40587))
        ((daysecond=uxsec%86400))
        ((hour=daysecond/3600))
        ((hoursecond=daysecond-(hour*3600)))
        ((minute=hoursecond/60))
        ((second=hoursecond%60))

#
#    Adjust things if we are negative
        if ((uxsec<0)) ; then
                ((mjd=mjd-1))
                ((hour=(hour+24)%24))
        fi

#
#    Convert mjd to year, month day and get dow (day of week)
        datecalc -j $mjd | read year month day
        dow=$(datecalc -D $year $month $day)

#
#   Format the date
        val=$hour
        time="${val}:"
        val=$minute
        time="${time}${val}:"
        val=$second
        time="${time}${val} $year"
        fdow=$dow
        val2=$day
        time="${fdow} ${months[month]} $val2 $time"
        echo "$time"
        return
}


integer unixsecond
typeset -R11 dsecond

while (($#)) ; do
        unixsecond=$1
        shift
        time1=$(unixsecond2timestring $unixsecond)
        dsecond=$unixsecond
        ((unixsecond2=unixsecond-(5*3600)))
        time2=$(unixsecond2timestring $unixsecond2)

        print "arg = $dsecond     ${time1}    ${time2}"
done
exit 0
```