

CnCalendar 历法说明

作者：刘啸 (liuxiao@cnpack.org)

部门：CnPack 开发组

类别：帮助文档

版本：V1.0.1.2

创建：2006.01.05

修改：2007.11.15

1、CnCalendar 介绍

CnCalendar 旨在为国内外程序员尤其是 Delphi/C++Builder 程序员提供一开放源码的历法计算工具包，此包涵盖了公历农历节气干支生肖星座数九三伏出梅入梅日出日落等各个方面，虽然力求做到完整精确，但历法本身的复杂性和天文现象的不确定性决定了这里的计算不会太完善。为了给用户一个完整的使用说明以及使用户在发现问题时能通过阅读 CnCalendar 的代码解决问题，这里将 CnCalendar 的历法算法说明整理如下文。

2、CnCalendar 算法说明

本节对 CnCalendar 所使用的算法进行了详细说明，欢迎阅读并指出谬误之处。

2.1 概念与常识介绍

进行历法计算前，有一些概念是需要首先明确的，包括历法本身的概念、部分天文名词、公历年、月、日、时的定义等，CnCalendar 算法中所涉及的概念都会在本节中加以说明。值得一提的是，部分概念是属于天文现象、部分则是历法推演而来，一般根据前者修正后者。前后两者拟合的精确度，是评判历法准确与否的依据。

2.1.1 阳历、阴历、阴阳历

太阳和月亮是对人类生产活动影响最大的两个天体，因此历法基本上都是以日和（或）月的运行规律为依据来计算的，纯粹以太阳的运行规律来计算的历法称为阳历，纯粹以月亮来计算的则称为阴历，两者兼而有之的自然就是阴阳历了。公历是典型的阳历，回历是阴历（本文未涉及），中国的农历则是阴阳历。

2.1.2 历理的依据

历法的作用有两个，一是能精确计时（或者说是计日），二是要符合实际情况。这个实际情况包括很多内容，如季节更替、农耕活动、潮汐涨落等。实际情况造就了历法，历法反过来又应当能根据其规则去推演实际情况，这便是历法的最终目的。目前通用的公历是 1582 年制定的格里高利历（Gregorian），它在前身儒略历（Julian）的一年 365 日、四年一闰的基础上增加了百年不闰、四百年又闰的规矩，形成了现在的通用公历历法。格里高利历是以数值计算为历理依据的，换句话

说，按照格里高利历的数值计算规则，可以往前后推任意多的天数而计算出当时的公元年月日来，至于是否符合实际情况，则只有到时候瞧着办了。像格里高利历的前身儒略历就出现过硬算下去误差积累导致春分日提前 10 天的事故，不过令人放心的是，太阳与地球的相对活动情况比较稳定，儒略历只是拟合得不甚精确，而修改后的格里高利历的精确度提高了许多，至少能支持三千年（如果加上四千年又不闰的规则则可以撑几万年了）。这也说明，只要历法所根据的天文情况比较固定，历法就可能推演成简单数值计算的公式。公历正是如此。

中国的农历计月所依据的是月亮的运行规则，月亮的不稳定度比太阳高得多，而农历是天文历法，其原则规定了以实际情况为准、数值计算为辅，所以导致了中国历史上的多次修历，更导致了目前农历计算和公农历转换的复杂性。而且农历这个阴阳历还必须考虑到太阳周期（回归年）和月亮周期（阴历月）的配合，更产生了十九年七闰的规则，而闰月的确定还需要以节气为准，大大增加了计算的复杂度。后文会详细说明。

2.1.3 春分点，回归年，阴历年

地球是斜着绕太阳转的，所以赤道面和黄道面有个固定的夹角，这个夹角在地球公转过程中导致了太阳直射点在南北回归线间移动，因此造就了四季更替、形成了年的概念。注意年的概念比发现地球绕太阳转要久远得多，年对古人的直观感受就是四季更替和春种秋收，这个感受和地球是否公转是无关的。假想如果地球光自转不公转而太阳在垂直于黄道面的方向上来回移动，同样也能造就年的感受。——说明这点的目的是为了区分下面所描述的概念：恒星年和回归年。

地球精确绕太阳转一周的时间称之为恒星年，而四季更替的精确时间称之为回归年。那么怎样衡量四季更替的精确时间？这就要用到春分点的概念。春分点是天文概念，它是黄道面上的地球公转轨道和赤道面的两交点之一（另一个自然是秋分点了），春分秋分点时太阳正好直射赤道，而太阳的直射又正好符合四季更替的感受要求，因此回归年的定义就是地球从这个春分点转一圈又回到春分点的时间。

如果春分点不动，恒星年就等于回归年，因为地球转一圈又回来了。遗憾的是地轴总会像陀螺一样有进动和章动，所以导致的岁差会让春分点在一年中稍微前进一点点，因此一恒星年比一回归年要长那么一点点，一个是 365.25636 日，一个是 365.2422 日。记住，回归年才是我们通常概念中的年。

也许有人要问，恒星年和回归年有 20 多分钟的差距，长此累积下去会有什么影响？答案是“和历法无关”。历法中本来就不管地球在公转轨道上的位置，春分点秋分点的平分——冬至点和夏至点也和公转轨道上的近日点远日点没什么必然关系，所以关心恒星年和回归年的差距是天文学家的事情。太阳历的基础就是回归年。

至于阴历年，则没有什么天文意义，只是月的集合而已，不过在农历这个阴阳历中，它也得拟合回归年的长度。一农历年是从正月初一到次年正月初一前的时间，它可能包含 12 或 13 个月，取决于闰年的设定。

农历中还有“岁”的概念，是从冬至日到次年冬至日之间的时间，等于一回归年。这个概念用来计算闰月。

2.1.4 阳历月，农历月，月朔

阳历月同样没有什么天文现象和它对应，设置阳历月只是为了便于把年分割计时，所以儒略历时能随便将月份的日期调大调小，只要保持一年的天数不变就行。像七八月连大、二月特小等都是历史上遗留下来的问题。

农历月则不同，农历的月是阴历月，反映了月相的变化，也就是月球绕地球公转一周的周期。农历规定了月朔时的那日为农历月首初一，下个月朔时为下个月月首初一。由于月球的公转周期差不多是 29.5 天但月必须是天的整数倍，因此农历月有大有小，大月 30 天，小月 29 天，而且也间隔得无规律可循，也就是说没法在不参考月相又脱离年份的情况下直接算出某个农历月的大小来，这一点和公历是不同的。

关于月朔也有说明：朔是日月黄经相等的时刻，但由于太阳和月亮运行的不均匀性，导致每两个相邻月朔之间的时间都不相等，中国唐初以前的农历采用的是经过长期观测统计求得的相邻月朔间的平均时间，将这个平均时间当成一朔望月，由此直接计算而来的各个月朔时刻称之为平朔。而唐初后农历中使用的修正后的比较精确的朔被称为定朔，它们之间是有相对误差的。

2.1.5 日，平太阳日

日是地球自转带来的昼夜交替，时、分、秒都脱胎于它。以往日的定义是真太阳日，也就是在地面上观测到的太阳视点绕地球一周又回到天球上天顶位置的时间间隔，这个间隔和地球精确自转一周的时间（恒星日）有点差别，因为地球还在公转，位置有点变化，其原理类似于春分日前进导致的回归年和恒星年的差距。另外地球的黄赤夹角以及椭圆轨道上的公转速度不均匀也导致了真太阳日的长短在一年中有所区别，为了消除这个差别，历法中引入了平太阳的概念。“平”在历法中是指时间均匀分布的意思，平太阳则是一个假想的太阳，以均匀的速度在天球赤道上绕地球运动，其速度的精确值为真太阳在一回归年内的平均速度，并且和真太阳同时经过近地远地点，这个平太阳运行一周回到天顶的时间则定义为一平太阳日，也就是现在通常的日的概念。换句话说，地球一年内的自转转速分布和公转情况共同决定了一平太阳日的实际大小。

至于时、分、秒，最初都是由平太阳日的定义衍生而来的：1 平太阳日等分为 24 小时，1 小时等分为 60 分钟，1 分钟等分为 60 秒。而后来由于地球自转不均匀导致的平太阳日的精确度不高，便另起炉灶以其他更精确的手段规定秒的长度，则是以后的事情了。

2.1.6 节气

节气是对一回归年内地球公转轨道上的 24 个具有季节意义的等分位置的描述，这 24 个节气名这里就不赘述了，只是这个“等分”也有说法：中国古代采用平气，按时间等分一回归年，相邻两节气之间时间间隔固定。清代修历后改用定气，等分周天为 24 分，每 15° 为一个节气，这样相邻节气的的时间间隔由于地球公转速度的变化会有所不同。

24 个节气中包括 12 个节气和 12 个中气，奇数号为节气，存在于每公历月上旬，偶数号为中气，存在于公历月下旬。中气对阴历的闰月计算起着决定性的作用。

本文的节气计算和中国现代农历一样使用定气。而古代使用的平气，其交节时刻和根据定气计算推得的时刻会有较大的误差，计算时应予注意。

2.1.7 年月日时的天干地支

中国采用天干地支计法来计年月日时，十天干十二地支循环配成六十花甲再轮排，年月日时都有和其对应的干支。计算年月日时的干支俗称“算八字”，当然本文的算八字是指根据年月日时推算出干支数，并非根据推算出来的四个干支再算命运财运桃花运等，那已经不属于本文讨论的范畴了。

3、 历法算法分类介绍

本节对 CnCalendar 中的历法算法进行了详细介绍，但部分算法属移植而来，暂时无法阐述。在说明算法之前，一些约定俗成的说法需要说明一下：

- 公元前 n 年表示为 $-n$ 年，公元 0 年不存在。
- 不存在公元 1582 年 10 月 5 日到 14 日这 10 天（被 Gregory 删掉了），但星期和干支仍然连续。
- 计算而得的干支、生肖、星座、节气的数字描述均以 0 为起始。
- 求回归年内第 N 个节气交节时刻时，第 1 个代表小寒，这里无第 0 个的说法。

CnCalendar 目前的历法算法包括：

- 星期
- 公历年月日时的干支
- 节气的公历交节时刻
- 星座
- 二十八宿日
- 公历转农历（部分）
- 数九三伏入梅出梅
- 日出日落时刻的计算

下面对各个算法分开进行简要说明。具体实现可以参考 CnCalendar 中的代码，里面也有较为详尽的注释。

3.1 计算星期

CnCalendar 提供了根据公历年月日计算星期几的方法，算法来源于《新编万年历》一书的公元日数求余法（有细微修改），原理很简单：星期的 7 天排列是和具体历法无关的，只按照日数循环计算，因此只要知道某个公历年月日距某个参照日的天数和此参照日的星期数，便可通过 7 求余法计算出公历年月日的星期几来。

算法中，距公元 1 年 1 月 0 日（公元前 1 年 12 月 31 日）的日数等于两部分之和：基础年日数和闰年修正日数。前者等于和公元 1 年的年份差乘以 365 再加上今年此日距今年 1 月 0 日的日数（包括此日本身，每个闰年里多出的 1 日也计算在内）。后者等于年份差对 4 求余再减去个修正

值。

年份差是该年和公元 1 年的实际年份差距，当 N 为正时，公元 N 年的年份差为 N - 1，公元前 N 年的年份差为 N（因为无公元 0 年）。

修正值是对儒略历和格里高利历切换期间的日期调整和对百年不闰四百年又闰规则的调整之和。其值规则如下：

- 日期为 1582 年 10 月 5 日之前含 5 日，以下同）时为 0。
- 日期为 1582 年 10 月 15 日至 1700 年 12 月 31 日时为 10。
- 从 1701 年 1 月 1 日起增加 1，每增加一个世纪再加 1，但能被 400 整除的世纪不累加。
如 1801 年元旦后修正值为 12，1901 年元旦后修正值为 13，2001 年元旦后修正值仍为 13。2101 年元旦后才为 14。

算出实际日期后，减去 2，再对 7 求余，得到的即是星期值，0 对应星期日，1 对应星期一，由此类推。这儿减去 2 是因为公元 1 年 1 月 2 日才是星期天（0 日是星期五，因此-2 也可用+5 代替）。

3.2 计算干支

计算年月日时干支的原理和计算星期差不多，因为其连续性，都可以用求余法来计算。不过中国用来计算干支的年月概念和公历年月不同，年以立春为界，月以 12 节气为界，计算前公历的年月需要转换成传统中国历法的年月。

3.2.1 年干支

计算年干支需要公历年月日三参数，无月日参数则默认认为是立春之后的某日，这样不需要调整年份。如有月日参数，则判断其是否处于本年立春交节时刻之前，之前则属于去年，年份减一。得到转换的年份后，假设为 Year，如其大于 0，则干支数为 $(AYear - 1 - 3) \bmod 60$ ，其中 Year-1 为到公元 1 年的年份差，Year 为负数则根据无公元 0 年调整为 $(AYear + 1 - 1 - 3) \bmod 60$ ，这个-3 源于公元 1 年是辛酉年，而辛酉在六十花甲的 0~59 中排行 57。

3.2.2 月干支

公历月份的干支数虽可以用求余法来计算，不过因为一年的月份数等于地支数，和天干数的关系也比较简单，因此有更简便的口诀算法（口诀来源于《新编万年历》一书，网上也有类似文章）。首先仍然需要根据当年的立春和各个节气交节时刻来调整年月数，然后计算出本年干数，年干数和本年首月月干数有口诀对应，这里把冗长的口诀简化了一下写在注释里，用代码描述如下：

```
case Gan of // 根据口诀从本年干数计算本年首月的干数
0,5: // 甲己 丙佐首,
    Result := 2;
1,6: // 乙庚 戊为头,
    Result := 4;
2,7: // 丙辛 寻庚起,
```

```
Result := 6;  
3,8: // 丁壬 壬位流,  
Result := 8;  
4,9: // 戊癸 甲好求  
Result := 0;  
end;
```

现行农历沿用了夏历，规定正月为寅。得到首月的干支数后，便可序次推得该月干支数。

3.2.3 日干支

日干支和星期的计算有类似之处，按星期计算的法子算得公元日数后，加 12 再对 60 取余则得到当日的干支数，12 源于公元 1 月 0 日是丙子日，而丙子在六十花甲的 0~59 中排行 12。

3.2.4 时干支

一天 24 小时对应着 12 个时辰，时辰和干支对应，但时辰的分界却不严格对应着日的分界。前一日 23 时到本日 1 时为子时，1 到 3 时为丑时，依此类推。

时干支和月干支的计算类似，首先根据小时数转成时辰数，如果是 23 时以后，则日期调整为后一天。接着计算出本日干数，本日干数和本日子时干数也有简单的对应关系，不过这次没找到口诀，大概因为对应比较简单。此对应关系用代码描述如下：

```
case Gan of  
0,5:  
Result := 0;  
1,6:  
Result := 2;  
2,7:  
Result := 4;  
3,8:  
Result := 6;  
4,9:  
Result := 8;  
end;
```

可以简化成：

```
if Gan >= 5 then  
Dec(Gan, 5);  
Gan := 2 * Gan;
```

所以不需要口诀了。

得到子时干数后，便可序次推得本日内时辰的干支数。

3.3 计算二十八宿

日二十八宿和星期的计算有类似之处，按星期计算的法子算得公元日数后，加 22 再对 28 取余则得到当日的二十八宿，22 源于公元 1 月 0 日是柳（柳土獐），而柳在二十八宿的 0~27 中排行 22。

3.4 计算节气

节气的定义虽然简单，可要计算出比较精确的交节时刻却并不容易，原因就是春分点在一回归年内会有细微移动，所以太阳两次通过各个定气点的相隔时间并不是一精确回归年，便没法以 365.2422 日为周期来直接计算各个节气的时刻。CnCalendar 的算法移植自中国日历类中作者根据曾次亮著作而拟合的节气算法，考虑到了地球自转的进动与章动，因此平气定气的精确度据作者说能达到 10 分钟左右。

3.5 计算星座

星座计算比较简单，仅仅根据月日就可以确定黄道十二宫的对应数值，规则如下：

01.20 - 02.18: 水瓶座

02.19 - 03.20: 双鱼座

03.21 - 04.20: 白羊座

04.21 - 05.20: 金牛座

05.21 - 06.21: 双子座

06.22 - 07.22: 巨蟹座

07.23 - 08.22: 狮子座

08.23 - 09.22: 处女座

09.23 - 10.22: 天秤座

10.23 - 11.22: 天蝎座

11.23 - 12.21: 射手座

12.22 - 01.19: 摩羯座

但黄道十二宫以白羊居首，所以代码中返回的星座序号中 0 对应着白羊，水瓶双鱼分别是 10、11。

另外星座名称有不同说法，如摩羯又称山羊、水瓶又称宝瓶，射手又称人马、天秤又称天平、处女又称室女等，这里只采用比较常用的说法。

3.6 计算农历

3.6.1 农历历法详细规则

中国的农历在 1929 年以前以北京当地的经度 $116^{\circ} 25'$ 为计算基准, 之后以北京时间也就是东经 120° 的东八区标准时间为计算基础。农历有四条基本规则:

1. 月朔时刻(定朔)所在之日定为农历月初一。
2. 如某节气交节时刻落于初一本日内(0 时后到 24 时前), 即使其交节时刻早于月朔时刻, 也算落在此月中。
3. 冬至必须出现于农历十一月内。
4. 相邻的冬至交接时刻间(岁内)如出现 13 次月朔也就是 12 个完整的农历月, 则此岁称为闰岁, 此岁中有一个月需要设置成闰月。因为期间必然有 12 个中气, 所以至少有一个农历月没有中气落入其中, 闰岁内第一个没有中气的月份定为闰月, 月份数与前一月相同。

对于第四条规则, 需要强调的是年内和岁内两个的概念是不同的, 如出现岁内有两个农历月无中气的情况, 则以第一个为闰月, 而此闰月可能不会落在本年内。而如果岁内有一农历月无中气但本岁只有 12 次月朔也就是 11 个完整的农历月的情况下, 那个无中气的月并不会被置闰。典型的例子是 2033 年闰 11 月: 2033 农历年第 8 个农历月和第 11 个农历月都无中气, 但因为 2032 农历年冬至(11 月)到 2033 农历年冬至(11 月)间只有 11 个完整的农历月, 所以不是闰岁, 无需加入闰月。2033 年冬至(11 月)到 2034 年冬至(11 月)间有 12 个完整的农历月, 所以是闰岁, 以冬至日后出现的第一个无中气的农历月为闰月, 正好是 11 月后面的一个农历月, 于是就闰 11 月。

总的说来一句话, 无中气只是闰月的必要条件并非充分条件, 只有岁内月朔太多需要置闰月时, 才会寻找岁内第一个无中气的作闰月。旧版历法以 2033 年第 8 农历月为闰 7 月, 只因为它是年内第一个无中气的农历月, 并未考虑到是否闰岁的情况。新版已经改为了闰 11 月。

另外, 由于目前节气采用定气, 地球在远日点时运动速度慢, 导致中气在冬天里推迟比夏日里慢, 因此无中气的月份大多都落在夏天前后, 几百年来极少有出现闰 12 月的, 这基本保证了春节的唯一。农历历法的精妙可见一斑。

注: 中国现代的历法以南京紫金山天文台发布的内容为权威。如本文的阐述和该权威有冲突, 以该权威为准。

3.6.2 农历计算

从上文的农历规则来看, 农历完全是天文历法, 加上月亮运动规律的不稳定, 导致很难对其进行精确计算。目前权威的计算方法是根据天文观测数据建立月球、地球、太阳的运行数学模型来精确计算各个朔日与中气的时刻再加以历法推算。

CnCalendar 中的公历转农历的函数移植自中国日历类的 Javascript 源码, 实现了公元前 850 年到公元后 2100 年的公历到农历的转换。但因为算法的复杂性, 此处无法详细阐明。

3.7 计算数九与三伏

数九也称九九，是从冬至日开始的 81 天，每 9 天称为一个九，第 10 天称为二九第一天，冬至日本身是一九第一天。CnCalendar 中计算某公历年月日的九九数时首先算出本年的冬至日，如果此日在冬至日开始后的 81 天内，则表明在九九日内，可通过整除求余等求出是几九的第几天。如果此日落在本日冬至日前，则检查是否处于上一年的冬至日的 81 天后，是则再计算一遍，因为九九日一般都会跨越两个公历年的交界时。

伏日的计算稍微复杂点，按传统，每年夏至后的第三个庚日（天干）起的 10 天为初伏，夏至后的第四个庚日起为中伏，如果第五个庚日落在立秋前则中伏算 20 天，否则 10 天，立秋后的第一个庚日起的 10 天为末伏。另外计算庚日时夏至立秋本身不计算在内，也就是说如果夏至日本日是庚日，它的后 10 天才算第一个庚日，立秋与此相同。

3.8 计算入梅出梅

入梅和出梅是指江南一带梅雨季节的开始和结束，入梅日规定为芒种之后的第一个丙日（天干），出梅日则是指小暑之后的第一个未日（地支），和上面的类似，计算日的天干地支数时芒种和小暑本日同样不计算在内。

3.9 计算日出日落

该部分算法移植自“时间科普网站”，此处无法详细阐明。

4、 参考内容与致谢

CnCalendar 的开发主要参考了以下内容，在这里对作者们的无私奉献表示敬意。

- 《新编万年历》，实际有百年左右，包括一些历法说明与计算。
- “知来者”日历源码以及“中国日历类”源码，JavaScript 编写，节气算法和公历转农历算法自此移植而来。
- 林洵贤的万年历 JavaScript 源码，参考。
- “日梭万年历” 5.2 beta 版，用来核对部分计算结果。
- “时间科普网站”：<http://www.time.ac.cn>，日出日落计算代码自此移植而来。

5、 联系我们

开发网站：<http://www.cnpack.org>

开发论坛：<http://bbs.cnpack.org>

管理员信箱：master@cnpack.org