美国夏威夷大学
分时系统的设计和实现

（查理·卡洛牙·贝斯）

ALOHA 系 統

上海计标所翻印

UHTSS —— 夏威夷大学分时系统的设计和实现

摘　要

1968年9月，夏威夷大学电子工程系开始了一个研究计划，ALOHA系统，以探索在一个地理上分散的计算机系统里使用普通商线通信的可能办法。当时在大学的IBM 360/65上还没有分时系统运转着，ALOHA系统便决定发起一项子计划，提供一个分时设备以满足它和大学两者的需要。经过对各种选择进行小心的探索之后，确定了没有现成的系统满足目前的要求和限制，因而研制了具有如下特点的系统 UHTSS。

1）不干扰正常的批次操作，

2）不依赖于OS/360的任何特殊文件；

3）利用OS/360的特色，

4）要求适量的内存作系统常驻部分并对另外每个活动用户要求少量额外的内存，

5）要求一定量的二级存贮器维持系统库并用作交换媒介

6）在一个大学环境里容易维持，那里被指定来进行设计的学生调转率是很高的；

7）提供一个基本的结构，从而语言翻译程序和其它用户设备只要花适量努力就可以结合进去，以及

8）可以在1至2年内设计，写出和实现。

UHTSS 用户可用的设备包括：

1）通过各种编辑和显示命令进行文件管理；

2）BASIC程序设计，

3）XPL程序设计（PL/I的一种方言）

4）台式计算器的操作方式；

5）作业的要记是360批次操作方式；

6) 远程调用成批操作的状态以及

7) 文件复制，联结，合并和重排序等应用程序。

本文记录了这些目标和设备是怎样达到的。

引　　言

一九六八年九月，夏威夷大学计算中心少数几个成员编写的一个分实分时系统问世了，这个分实分时系统不触与IBM系统/360操作系统，OS/360的新版本同时工作。这种对基本操作系统设计稳定性的依赖性已证明是一个在其它方面考虑都很周到的系统的致命伤。随着每一个OS/360的更新，就得改变内部机制，这种前景是计算中心所不能容忍的。所以当校方刚要开始对分时系统的重要性进行估价时，学校的分时系统就被搁置起来了。

然而，一九六八年九月，夏威夷大学电气工程系和信息科学课程却接到了美国国防部THEMIS计划的一个研究合同。这项计划即ALOHA系统是用来研究一个在地理上很分散的计算机系统掌握用通常有线通讯的可能性的[1]。当然，ALOHA系统需要一个中央分时设备用来为其远端用户队列服务。因为没有分时系统运转着，ALOHA系统决必须启一个分计划。这个分计划提供的分时设备既可满足ALOHA的需要，也可满足学校的需要。在仔细地研究了各种可取的方案以后，决定研制一种新的系统而不是採用现存的系统，这个系统称为夏威夷大学分时系统或UHTSS。

决定研制UHTSS而不是採用现存系统的根据是没有既满足ALOHA系统的要求，又满足学校计算机资源限制的分时设备。那时，在360上运行的许多分时系统是一种独立系统。它们完全独占了硬设备而排除成批处理。学校只有一台360，而成批处理却是绝对必要的一种操作。因为所有的现存用户都是面向成批处理的，他们的卡片。带和盘都是OS/360格式。因

此，严重地割裂成批操作的系统是不能被认真地考虑的。这的确是一种严重的限制，ALOHA 系统要扩大学校的程序计算能力，又要不对360的操作承担责任。所有这些实质上意味着不得不保留OS/360。

这种不排除OS/360的军兜的系统需要非分时系统专用的内存储器和外存储器。当时360结构包含有一台512K（1K＝1024）字节的内存储器和13台IBM 2314磁盘驱动装置，每台容量约为29×10 字节。计算中心非正式地赞成分配200K字节内存和一台2314 磁盘驱动装置给分时系统。在这种苛刻的限制下，没有任何现存系统可以使用。

对于研制UHTSS 来说除了实际的理由以外，而且能从这种冒险事业中获得试验研究机会也是有趣的。

这样，一九六九年一月，由三个毕业生包括作者在内，在当时任计算中心负责人W．W．Peterson 博士的指导下，非正式地开始了UHTSS 设计。几个月后，经过设计，大约在一九六九年六月开始实际编写程序。编写和调试大约一年以后，一九七〇年八月系统进入运行。从一九七〇年九月直到一九七二年一月，UHTSS 满足了ALOHA 系统和学校主要部分的分时需要。一九七二年一月採用了IBM的TSO（Time－Sharing Option）系统〔2〕作为学校的基本分时设备，废弃了UHTSS。不管对这个行动是赞成还是反对，UHTSS还是达到了它的目的，在没有现正可採用的方法时，提供了分时系统，并提供了有价值的经验和研究机会。

本论文是这个断言的主要例子。作者以为本文说明了一个有意义的软件设计及其实现方案，这对于许多计算机科学领域，特别是分时的一般理解和改进是有益的。

写在前面

计算机事业的迅速发展推动了计算机网络工程的研究，在我国，这项工作正在进行。由于工作上的需要，我们曾查阅过一些资料。同时，我所721大学的英语班在结业前要选一些英文资料作为翻译实习，于是选中了本书，组织英语班的部分学员集体翻译，参加本书翻译的同志是：

卢致皓　（引言）
吴　敬　（第1章）
叶蜀达　（第2章）
徐英凤　（第3章　第6章）
刘东良　（第4章，第10章）
胡竞瑾　（第5章，第6章，第7章）
吴　煒　（第7章）
林斌贵　（第9章）
吴文祺　（第11章）
张锡平　（第12章）

负责组织和校对的同志是：夏慕修　程惟守　杨人鹤　徐克仁

由于本书的翻译是习作，缺点错误一定不少，请批评指正。

上海市计算技术研究所
情报资料室

目　录

第一章　目　的

分时是一门使得多用户在他们各自的远程终端上同时地存取一台计算机资源的技术。尽管许多终端能同时地使用计算机，但系统努力提供一种服务方式，使得每个终端操作员感觉到唯有他独占计算机。自从麻省理工学院的一组计算机科学家于五十年代末期首次提出这种概念以来，许多实现证明了这种想法是可行的，并且对各种计算机应用也是有用的［3，4，5，6，］。

大多数分时系统的理论基础是为共同简短而高度相互作用程序的用户提供快速服务，以方便调试，促进计算方法的试验与改进，并允许把计算机当成一台完善的台式计算器来使用［7］。这也就是UHTSS研制的指导原则。要感到大学分时系统的目的不是为此用户可以利用的全部OS／360 能力提供远程存取。为了保持简单性并获得速度和效率，UHTSS与用户之间的接口，即系统命令，UHTSS与360 操作系统之间的接口限于那些支持所述分时概念的特色。

这种接口的简单性由于消除了尽可能多与实际问题解决过程无关的细节，如资源的分配与说明，也会增加交给广大用户应用的可能性。为实行和满足广大用户普遍应用，同等重要的是用于解决问题的工具或语言，BASIC 由于其简单，实用和流行而被选择作为这种工具。

因为远程终端的获得大都是用户个人的事情，取决于用户的选择，分时设备就必须灵活得对可获服务的各种终端都是实用的。在UHTSS的研究阶段中，唯一可用的终端是 IBM

2260 ，一台具有局部更新和字母数字显示能力的 CRT 型终端。ALOHA 系统计划首先通过无线电联系将电传终端联到一台 Hewlett－Packard 2115A小型计算机，该小型计算机再通过它的一个选择通道与 360 联接。同时发现了 IBM 将在大学团体中增进其 2741 型终端打字机。为了满足这个可能的设备配置的各种要求，UHTSS必须对终端增加扩充提供接口能力。

考虑到研究这些因素，决定研制这样一个分时系统：

1. 不干扰正常的成批处理；

2. 不依赖于 OS/360任何特殊版本；

3. 利用 OS/360的特色；

4. 需要适量的存贮器放系统常驻部分，并对每个另外的活动用户要求少量额外的存贮器；

5. 需要一定量的二级存贮器来保留系统库并用作交换媒介；

6. 在大学环境中易于维持，在那里被指定来进行设计的学生周转率是很高的；

7. 提供一个基本结构，以而语言翻译和其它用户设备只要花适量努力就可以结合进去；

8. 以及 3至4 人能在一至二年内设计，写出与实现。

这篇论文记录了这些目标是怎样达到的。

第二章　邏輯設計

2.1. 功能的定义：

一个分时系统的逻辑功能可以定义为：

(1)　系统活动的管理

(2)　用户和系统文件的管理

(3)　用户程序执行的管理，和

(4)　终端输入／输出管理(8)。

代表这些逻辑功能的模块职责如下：

2.2. 系统活动的管理：

提供这个功能的模块——TMON，监控UHTSS各部分之间的通讯流，每当用户操作或系统内部需要服务时，就通知TMON，然后TMON决定哪一個模块被分配给此未决任务，并且在适当模块的分配队列中置一登记项，之后TMON接到该模块的通知，分配任务完成，这时它撤销排队中的登记项，并决定有无其它操作序列要启动。

图2.1举例说明UHTSS不同模块之间的通讯流，TMON通过特有的队列把任务指定给其它模块，这些模块是直接与TMON进行通讯，为此，TMON具有最高的执行优先权。因此每当它请求中央处理机(CPU)时，它就得到控制，然后它完成请求任务并放弃控制，TMON并不执行输入／输出或任何其它在完成前需要挂起一个任务的操作，系统活动管理是一個线性过程，每一個任务都是按照请求，在下一個任务能被分配之前，启动和完成，不需要一个队列，而所有其它模块，都可以在

任何时刻被它中断和自动像输入/输出那样的操作，这要求让出中央处理机，因此可以按与完成任务无关的速度对它分配任务，并要有一个队列管理任务分配。

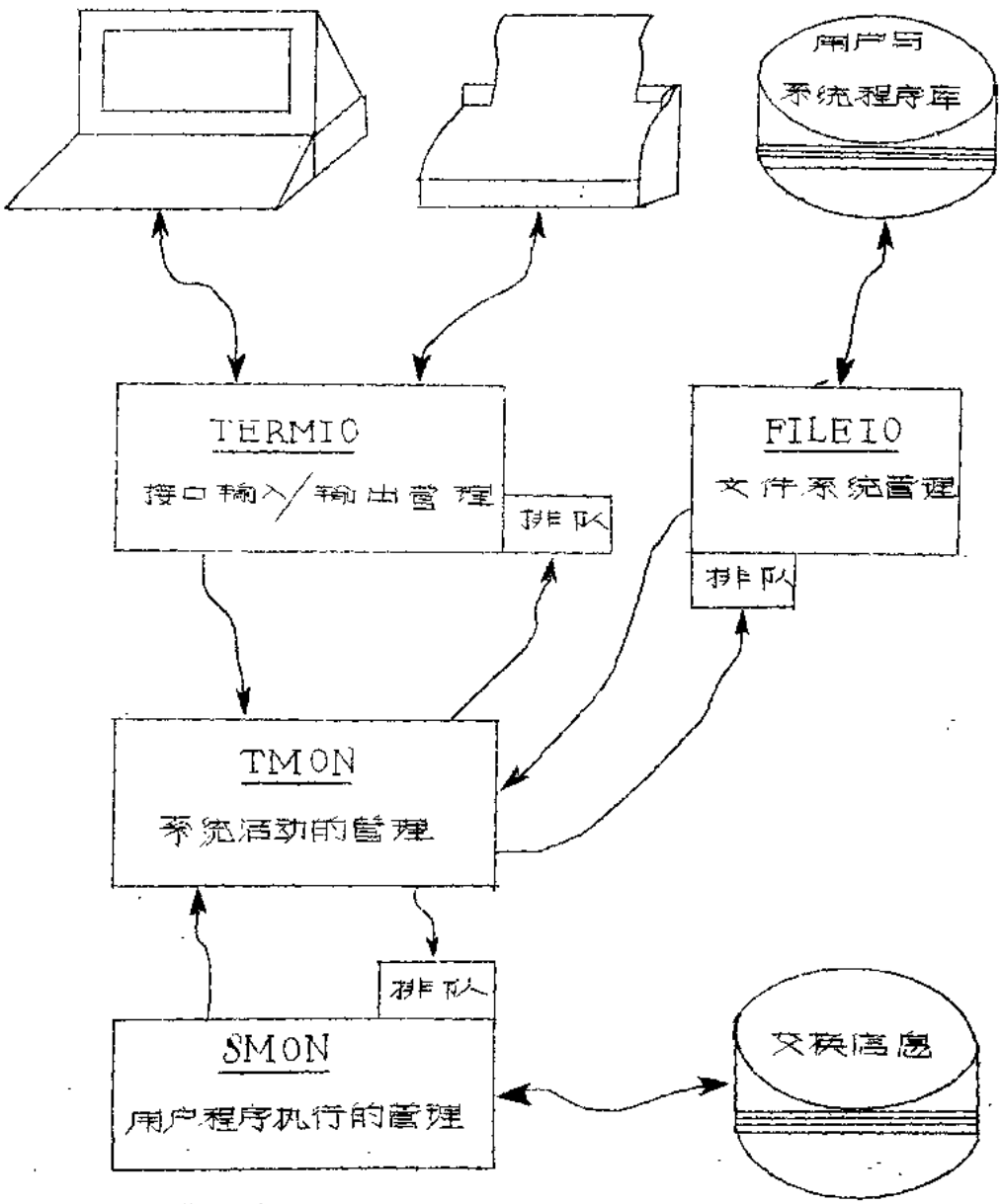

图2.1 逻辑模块组件之间的通讯流

每個活动终端的状态，包括未决请求的详细描述，保存在一张称为系统应用者文件（System User File）或SURF 的表中。SURF 的详细叙述反其应用见表 2.1 和表 2.2。除了状态参数以外，每一個 SURF 登记项包括一個 80 個字符的缓冲器，用来进行终端和系统内部消息的处理。所有的终端和系统内在通讯的标准记录长度是 80 個字符，由于它可被所有 UHTSS 系统模块存取，以及它又是模块组件之间通讯的主要机制，這個 SURF 表是整個系统活动的焦点。

通过调整 SURF 表中的单元和维持各种模块的队列，TMON 控制系统内部操作，并因此而控制 UHTSS 的性能。第四章是 TMON的详细叙述。

SURF 是系统用户的文件，对於系统中每一個活动终端，在 SURF 中有 65 個字登记项。各 SURF 登记项内部信息区如下：

区域名称	区域长度 (72 bit字)	对起始点的偏移
TERMNL	1	0
TYPE	1	1
JOB	2	2
NAME	5	4
STATUS	1	9
BUFFER	20	10
IOSTAT	1	30
IOTYP	1	31
UNUSED	1	32
SMONTYPE	1	33
POINTERT	1	34
POINTERF	1	35
POINTERS	1	36
REPLY	1	37
ORIGIN	1	38
BLOKSTAT	1	39
CONDCODE	1	40
IOCOUNT	1	41
AREALOC	1	42
DSHEAD	1	43
DSPNTB	1	44
DEBUG	1	45
SDSPNTR	1	46
PARM2	1	47
CDSPNTR	1	48
UNUSED	1	49
TIMER	1	50
OSTAT	1	51
FILEPNTR	12	52
UNUSED	1	64

表 2.1　　SURF 表细目

表 2.1 SURF参数细目

TERMNL — 顺序终端标识数　　从 01 起始增量 01。

TYPE　—00 — IBM 2260　(注1)
　　01 — 电传打字机　　(终端机型)
　　02 — IBM 2741　(注2)

JOB — 现行用户的帐号。

NAME — 现行用户名称。

STATUS —00 — 终端空。
　　01 — 终端通讯有问题。
　　02 — 文件 输入/输出 请求未定
　　03 — 等候到文件 输入/输出 的输入
　　04 — 不用。
　　05 — 执行请求未定。
　　06 — 活动 — 在交换区域执行。
　　07 — 活动 — 等候到执行。
　　08 — 不活动 — 等候输入。
　　09 — 不活动 — 等候输出。
　　10 — 不活动 — 等候的信息选择。
　　11 — 不活动 — 应用户请求。
　　12 — 全部离线止。

BUFFER — 终端 输入/输出 缓冲器 — 30 个字节。

IOSTAT —00 — 空闲状态。
　　01 — 输入/输出请求未定。
　　02 — 文件 输入/输出 在用缓冲器。
　　04 — SMON 在使用缓冲器。

(注1) 这是 360 系统中一种显示设备型号。

(注2) 这是 360 系统中一种控制台打字机型号。

IOTYPE = 终端 输入/输出 模块的请求代码

00 = 输入操作完成

01 = 输出和释放磁盘 — 最后一行

02 = 全屏写中

03 = 输出且不释放磁盘

04 = 输出完成

05 = 通知文件 输入/输出 缓冲器可利用

06 = 来自 SMON 队列的远程请求

07 = 通知 SMON 缓冲器可利用

08 = 输出退出消息

09 = 文件 输入/输出 完成

10 = 选择

11 = 创造

12 = 停止未决的请求

13 = 用户中断

14 = 名称

15 = 打印

16 = 穿孔

17 = 显示

18 = 插入

19 = 删去

20 = 涂去

21 = HASP （注 3）

22 = 分配

23 = 自由

24 = 不用

25 = 不用

（注 3）这是 OS/360 一个 I/O SPOOLing 系统

26 = DAOL

27 = 不用

28 = 连接

29 = 装入

30 = SMON 请求完成文件输入/输出

31 = 请求交换监控程序

32 = SMON 中程序输入完成

33 = SMON 中程序输出完成

34 = 终端断开

35 = 终止 SMON 中的作业

SMONTYPE = SMON 提供请求代码，
代码在 IOTYPE 区域描述

POINTERT = 在终端 输入/输出 排队中，下一个 SURF 入口的相对地址。

POINTERF = 在文件 输入/输出 排队中，下一个 SURF 入口的相对地址。

POINTERS = 在 SMON 排队中，下一个 SURF 入口的相对地址。

REPLY = 下列代码标识的系统提供在等待序取 SURF 缓冲区。

0 = 没有未决的请求

2 = 文件输入/输出

4 = SMON

ORIGIN = 0 = 终端 输入/输出 或文件 输入/输出 引起请求。

2 = SMON 引起请求，但是不要响铃。

4 = 交换监控程序产生的请求

6 = 在交换区域内，用户发出的内部请求。

8 = 在交换区域内，用户发出的终端输入/输出。

BLOKSTAT = 如果非另，则未决的 输入/输出 请求正由终端 输入/输出 模块起动。

CONDCODE = 文件 输入/输出 请求完成的状态。

0 = 操作成功地完成。

1 = 在操作中，检测出文件到端。

2 = 在操作期间硬设备出错。

3 = 非法操作请求。

4 = 合法请求不能完成。

IOCOUNT = 回引进后，输入/输出引用的数目。

AREALOC = 用户缓冲区绝对地址。

DSHEAD = 头数据组节（Head Data Set Node）的索引。

DSPNTR = 现行数据组节（Current Data Set Node）的索引。

DEBUG = 0 = 非调试。

-1 = 用系统模块跟踪活动进行调试。

TIMER = 自从上机后，所用CPU时间，以秒计。

OSTAT = 起起前的程序状态。

FILEPNTR = 定义数据组节的12个指示器。

-1 = 终端输入/输出。

-2 = 不定义。