Reading Usenet News with tin

This handout will introduce you to reading usenet news with *tin*(1). Usenet news is similar to an electronic bulletin board; users across the network post messages for others to read, asking for discussion, answers to questions, *etc.* Messages typically consist of text entered by users, but messages may also contain encoded binary files, such as executable programs, pictures, or compressed files such as *tar*(1) files. The range of categories available open to discussion includes the gamut from want ads to discussions about bald spots. We will use the newsgroups to post information regarding the class. The newsgroup *ucd.class.ecs40* is a read-only newsgroup where the teaching assistants, grader, or I will post important information. The newsgroup *ucd.class.ecs40.d* is a discussion newsgroup where you may post information to discuss class materials with your peers, or ask us questions which you think others might be interested in.

The original program to read news was called *readnews*(1). Since its inception, a plethora of other programs have been created which improve upon the original readnews, for example, *rn*(1), *xrn*(1), *trn*(1), *nn*(1), or *frn*(1). Most of these readers enhance the original by providing better interfaces and threading among followup messages. This handout will describe the basics of a threaded newsreader called *tin*(1).

To start tin, type

```
tin
```

If this is the first time you have used tin, you will get some introductory text as follows:

```
dec32> tin
tin 1.2 PL1 [UNIX] (c) Copyright 1991-93 Iain Lea.
Connecting to news.ucdavis.edu...

                 *****************************************
                 *      Welcome to news.ucdavis.edu      *
                 *****************************************

On the Information Resources computers:
Tin now uses pico as the default editor.  If you wish to use vi as your
editor in tin, run 'tin-vi' at the Unix prompt.

If you are a faculty member and would like to set up newsgroups for class
use, mail usenet@ucdavis.edu. (or other campus groups, clubs, etc.)

If you have questions about the use of tin 1.2, you can mail
ithelp@ucdavis.edu, or call the IT-CAP at 2-2548.  Any problems regarding
the functionality of tin should be reported to usenet@ucdavis.edu

Press any key to continue...
Reading news active file...
Reading attributes file...
Reading newsgroups file...

               tin 1.2 PL1 [UNIX] (c) Copyright 1991-93 Iain Lea.

Welcome to tin, a full screen threaded Netnews reader. It can read news locally
(ie. <spool>/news) or remotely (-r option) from a NNTP  (Network News Transport
Protocol) server. tin -h lists the available command line options.

Tin has five  newsreading levels,  the newsgroup  selection page,  the spooldir
selection page,  the group index page,  the thread listing page and the article
```

```
viewer. Help is available at each level by pressing the 'h' command.

Move up/down by using the terminal arrow keys or 'j' and 'k'.  Use PgUp/PgDn or
Ctrl-U and Ctrl-D to page up/down. Enter a newsgroup by pressing RETURN/TAB.

Articles, threads, tagged articles or articles matching a pattern can be mailed
('m' command), printed ('o' command), saved ('s' command), piped ('|' command).
Use the 'w' command  to post  a news  article,  the 'f'/'F' commands to  post a
follow-up  to  an existing  news article and the 'r'/'R' commands to  reply via
mail to an existing news articles author.  The 'M' command allows the operation
of tin to be configured via a menu.

For more information read the manual page, README, INSTALL, TODO and FTP files.
Please send bug reports/comments to the programs author with the 'R' command.
```

Next, you will be presented with a menu similar to that shown below. The first column indicates the number of each newsgroup, the second column shows the number of new messages in that group. The third column holds the actual name of the newsgroup while the last column gives a brief description of what the newsgroup is about. Use the arrow keys to move the cursor back and forth to select which group to read. At the bottom of the screen is a help menu which lists commonly used commands. Type **H** for a more complete help screen. Use **PgUp** or **PgDn** to move by a whole screen at a time. You may also use *vi*(1) commands to move; **K** up, **J** down, **^F** to scroll forward one screen, and **^B** to scroll backwards one screen.

```
1      188   ucd.forsale           Things for sale (offers and requests)
2      180   ucd.general           Items of general interest to UCD and D
3      18    ucd.jobs              Jobs offered and wanted
4      402   ucd.life              Davis Chatter
5      23    ucd.housing           Rentals available in Davis
6      113   ucd.itcap             IT-CAP announcements
7      287   ucd.test              Post test messages here
8      5783  ba.general            Announcements of general interest to a
9      6504  ba.jobs.offered       Job Postings in the Bay Area.
10     1828  ca.general            Of general interest to readers in Calf
11     28937 misc.forsale          Short, tasteful postings about items f
12     7684  rec.bicycles.misc     General discussion of bicycling.
13     23817 rec.humor             Jokes and the like.  May be somewhat o
14     3565  rec.pets              Pets, pet care, and household animals

       <n>=set current to n, TAB=next unread, /=search pattern, c)atchup,
    g)oto, j=line down, k=line up, h)elp, m)ove, q)uit, r=toggle all/unread,
      s)ubscribe, S)ub pattern, u)nsubscribe, U)nsub pattern, y)ank in/out
```

To add a new newsgroup, use the goto command. After you type **g**, *tin* will prompt you for the newsgroup to go to. To subscribe to the newsgroup used in this class, enter *ucd.class.ecs40*. After determining the newsgroup, *tin* will ask you where to put the new group in the list. Type **1** to put it first in the list of newsgroups.

```
Goto newsgroup []> ucd.class.ecs40
Position ucd.class.ecs40 in group list (1,2,..$) [18]> 1
```

The newsgroup will be placed into position 1 of your list. Your main menu will now look something like:

```
1      7     ucd.class.ecs40.sec01 Intro Software Development
2      188   ucd.forsale           Things for sale (offers and requests)
3      180   ucd.general           Items of general interest to UCD and D
4      18    ucd.jobs              Jobs offered and wanted
5      402   ucd.life              Davis Chatter
6      23    ucd.housing           Rentals available in Davis
7      113   ucd.itcap             IT-CAP announcements
```

```
8      287   ucd.test              Post test messages here
9      5783  ba.general            Announcements of general interest to a
10     6504  ba.jobs.offered       Job Postings in the Bay Area.
11     1828  ca.general            Of general interest to readers in Calf
12     28937 misc.forsale          Short, tasteful postings about items f
13     7684  rec.bicycles.misc     General discussion of bicycling.
14     23817 rec.humor             Jokes and the like.  May be somewhat o

       <n>=set current to n, TAB=next unread, /=search pattern, c)atchup,
    g)oto, j=line down, k=line up, h)elp, m)ove, q)uit, r=toggle all/unread,
      s)ubscribe, S)ub pattern, u)nsubscribe, U)nsub pattern, y)ank in/out
```

You should also subscribe to the newsgroup *ucd.class.ecs40.d*, the discussion newsgroup for the course.

To read the messages in a newsgroup, move the cursor onto the newsgroup and press enter. You will be presented with a new menu similar to the one shown below. As always, a help menu is available at the bottom of the screen, and you may use the arrows to move and select which article to read. New articles are denoted by a **+** sign. If a number is next to an article, it indicates how many other articles are in that "thread." In this example, one article has been posted in reply to the article from Michelle Hoyle entitled "Comments on Assignment 1".

```
           ucd.class.ecs40.d (3T 1A 0K 0H)      h=help

      1     +      This Newsgroup               Matt Bishop
      2     +1     Comments on Assignment 1     Michelle Hoyle

    <n>=set current to n, TAB=next unread, /=search pattern, ^K)ill/select,
  a)uthor search, c)atchup, j=line down, k=line up, K=mark read, l)ist thread,
    |=pipe, m)ail, o=print, q)uit, r=toggle all/unread, s)ave, t)ag, w=post
```

If you select the first article and press the enter key, it will be displayed as follows:

```
Thu, 06 Jan 1994 10:26:40   ucd.class.ecs40.sec01.d        Thread    1 of    1
Lines 10                       This Newsgroup              No responses
bishop@nob.cs.ucdavis.edu  Matt Bishop at Department of Computer Science, U

Just a reminder and a post to get things rolling on this newsgroups.
This group is where you, the students, may post and discuss the class
assignments or any problems/comments you may have.

Hope this helps,
Matt Bishop

    <n>=set current to n, TAB=next unread, /=search pattern, ^K)ill/select,
        a)uthor search, B)ody search, c)atchup, f)ollowup, K=mark read,
        |=pipe, m)ail, o=print, q)uit, r)eply mail, s)ave, t)ag, w=post
```

Once again, useful commands are shown at the bottom of the screen. Press the tab or enter key to read the next message in the thread (if any). Some of the more useful commands are summarized below:

o Print the article on the printer
m Mail this article to an email address
r Reply mail. Send mail to the poster of this article, quoting from the article
w Post a new article unrelated to this article
f Post a followup article related to this article
s Save article to disk. You have the option of saving the article directly, or automatically extracting encoded files (uuencode, shar, or zip formats).

As an example of posting a followup article, by pressing **f** you will be placed into the editor with the contents of the previous article "quoted" (i.e. preceded by ":" characters). This is useful in order to maintain the context of the previous article so that readers of your post are reminded of what you are replying to.

```
Subject: Re: This Newsgroup
Newsgroups: ucd.class.ecs40.sec01.d
References: <CJ80KH.5Gz@ucdavis.edu>

Matt Bishop (bishop@nob.cs.ucdavis.edu) wrote:
: Just a reminder and a post to get things rolling on this newsgroups.
: This group is where you, the students, may post and discuss the class
: assignments or any problems/comments you may have.

Hello, I'd like you to clarify part XXX in assignment ZZZ.
Thank you,
Diligent but confused ecs40 student.


Check Prepared Article
Your article will be posted to the following newsgroup:
           ucd.class.ecs40.d      Intro Software Development (dis)
```

This should be enough to get you started using netnews. Some frequently asked questions follow:

Q: Typing **H** doesn't bring up a help screen, but instead removed the miniature help screen from the bottom of the window.
A: **H** toggles whether or not to show the help screen at the bottom of each window. Use **h** to get a listing of the help screen. In general, uppercase and lowercase commands do different things.

Q: After reading the messages, they don't show up anymore and I want to read them again.
A: By default, tin only displays new articles. From the article menu, type **r** to toggle between displaying all articles or only new articles.

Q: There are hundreds of new messages I don't want to see. How do I reset the number of new messages to zero?
A: Type **c** to catch up and mark all articles as read. This is useful if you are subscribing to a new newsgroup with lots of messages or if a large number of messages have collected over time.

Q: An old article seems to be gone. How can I get it back?
A: You can't retrieve extremely old articles. Articles are automatically deleted by the system after approximately two weeks. Consequently, you should check news frequently and be sure to save any articles you think are important.

This handout has only covered a few of the basic aspects of *tin* and netnews. The best way to learn more is to simply play and experiment. Take advantage of the help screens, and feel free to subscribe to newsgroups which interest you.

Acknowledgement

Kendrick Mock wrote this handout, and Matt Bishop modified it slightly.

Sample Basic C Programs

Program #1: Fahrenheit and Celsius, Version 1

This prints a table with two columns, the left being Fahrenheit degrees, and the right the corrsponding Celsius temperatures.

```
#include <stdio.h>

/*
 * print a table for Fahrenheit to Celsius
 * from 0 F to 300 F
 */
void main(void)
{
      int fahr;                    /* fahrenheit temperature */
      int celsius         ;        /* celsius temperature */
      register int lower = 0;      /* begin table here */
      register int upper = 300;    /* end table here */
      register int step = 20;      /* increment */

      /*
       * print out the lines for the table
       */
      fahr = lower;
      while(fahr <= upper){
            /* get corresponding temp in degrees C */
            celsius = 5 * (fahr - 32) / 9;
            /* print it */
            printf("%d\t%d\n", fahr, celsius);
            fahr += step;
      }

      /*
       * say goodbye
       */
      exit(0);
}
```

Program #2: Fahrenheit and Celsius, Version 2

This does the same thing using different control structures and floating point arithmetic.

```
#include <stdio.h>

#define LOWER      0             /* begin table here */
#define UPPER      300           /* end table here */
#define STEP       20            /* increment */

/*
 * print a table for Fahrenheit to Celsius
 * from 0 F to 300 F
 * floating point version
 */
void main(void)
{
      float fahr;         /* fahrenheit temperature */

      /*
       * print out the lines for the table
       */
      for(fahr = LOWER; fahr <= UPPER; fahr += STEP)
           printf("%3.0f\t%6.1f\n", fahr, (5.0/9.0) *
                                                (fahr - 32));

      /*
       * say goodbye
       */
      exit(0);
}
```

Program #3: Version 1 of *cat*(1)

This short program copies the standard input to the standard output.

```
#include <stdio.h>

/*
 * copy input to output: short version
 */
void main(void)
{
	int c;			/* input character */

	/*
	 * copy the input to the output
	 * one char at a time
	 */
	while ((c = getchar()) != EOF)
		putchar(c);

	/*
	 * say goodbye
	 */
	exit(0);
}
```

Program #4: Version 2 of *cat*(1)

This does the same thing, again uses different control structures. Most C programmers would write this program the previous way.

```
#include <stdio.h>

/*
 * copy input to output: long version
 */
void main(void)
{
	int c;			/* input character */

	/*
	 * copy the input to the output
	 * one char at a time
	 */
	do {
		/* read a char */
		c = getchar();
		/* write a char (unless it's */
		/* the end of file marker)   */
		if (c != EOF)
			putchar(c);
	} while (c != EOF);

	/*
	 * say goodbye
	 */
	exit(0);
}
```

Program #5: A Version of *wc*(1)

This program counts the number of lines, words, and characters in its input, where a "word" is defined as any sequence of non-whitespace characters (blanks, tabs, and newlines).

```
#include <stdio.h>

#define IN_WORD           1      /* currently inside a word */
#define NOTIN_WORD        0      /* currently not in a word */

/*
 * count the number of lines, words, and chars in the input
 * a word is a maximal sequence of nonspace characters, so
 * the quote "+++ --- hi bye 879+3" has 5 words ("+++", "---",
 * "hi", "bye", and "879+3")
 */
void main(void)
{
	register int c;            /* input char */
	register int nl;           /* line count */
	register int nw;           /* word count */
	register int nc;           /* char count */
	register int state;        /* in or not in a word? */

	/*
	 * initialize
	 */
	nl = nw = nc = 0;
	state = NOTIN_WORD;

	/*
	 * handle input a char at a time
	 */
	while((c = getchar()) != EOF){
		/* got another character */
		nc++;
		/* is it a newline? */
		if (c == '\n')
			nl++;
		/* is it a word separator? */
		if (c == ' ' || c == '\t' || c == '\n')
			/* YES -- change state */
			state = NOTIN_WORD;
		else if (state == NOTIN_WORD){
			/* NO -- we're now in a word; update */
			/* the counter and state if need be  */
			state = IN_WORD;
			nw++;
		}
	}

	/*
	 * announce the results and quit
	 */
	printf("%6d\t%6d\t%6d\n", nl, nw, nc);
	exit(0);
}
```

Program #6: Digit and Space Counter, Version 1

This is similar to the word counting program, but counts digits, whitespace, and everything else (grouped into the category "other"). Note the array.

```
#include <stdio.h>

/*
 * count the number of each digit, whitespace,
 * and all other chars
 */
void main(void)
{
	register int c;			/* input char */
	register int nwhite = 0;		/* whitespace count */
	register int nother = 0;		/* other count */
	register int i;			/* counter in a for loop */
	int ndigit[10];			/* digit counts */

	/*
	 * initialize the ndigit array
	 */
	for(i = 0; i < 10; i++)
		ndigit[i] = 0;

	/*
	 * handle input a char at a time
	 */
	while((c = getchar()) != EOF){
		/* see what it is */
		if (c >= '0' && c <= '9'){
			/* it's a digit -- bump the right count */
			ndigit[c - '0']++;
		}
		else if (c == ' ' || c == '\t' || c == '\n'){
			/* it's whitespace */
			nwhite++;
		}
		else{
			/* it's neither a digit nor whitespace */
			nother++;
		}
	}

	/*
	 * announce the results and quit
	 */
	printf("digits: ");
	for(i = 0; i < 10; i++){
		printf("'%c' %3d\t", i + '0', ndigit[i]);
		/* put 5 digits per line, for neat output */
		if (i == 4)
			printf("\n        ");
	}
	putchar('\n');
	printf("whitespace: %d\nother:      %d\n", nwhite, nother);
	exit(0);
}
```

Program #7: Digit and Space Counter, Version 2

This is also a digit counter, but uses a `switch` statement rather than an `if … else if … else` statement.

```
#include <stdio.h>

/*
 * count the number of each digit, whitespace,
 * and all other chars
 */
void main(void)
{
	register int c;			/* input char */
	register int nwhite = 0;		/* whitespace count */
	register int nother = 0;		/* other count */
	register int i;			/* counter in a for loop */
	int ndigit[10];			/* digit counts */

	/*
	 * initialize the ndigit array
	 */
	for(i = 0; i < 10; i++)
		ndigit[i] = 0;

	/*
	 * handle input a char at a time
	 */
	while((c = getchar()) != EOF){
		/* see what it is */
		switch(c){
		case '0': case '1': case '2': case '3':	/* digit */
		case '4': case '5': case '6': case '7':
		case '8': case '9':
			ndigit[c - '0']++;
			break;
		case ' ': case '\t': case '\n':		/* whitespace */
			nwhite++;
			break;
		default:		/* neither a digit nor whitespace */
			nother++;
			break;
		}
	}

	/*
	 * announce the results and quit
	 */
	printf("digits: ");
	for(i = 0; i < 10; i++){
		printf("'%c' %3d\t", i + '0', ndigit[i]);
		/* put 5 digits per line, for neat output */
		if (i == 4)  printf("\n        ");
	}
	putchar('\n');
	printf("whitespace: %d\nother:      %d\n", nwhite, nother);
	exit(0);
}
```

Program #8a: Powers of 2 and –3

This program prints a table of 2 and -3 raised to the powers 0 to 9 inclusive. It illustrates the use of function calls.

```
#include <stdio.h>

/*
 * prototype (forward declaration)
 */
int power(int m, int n);

/*
 * generate a table of powers of 2
 */
void main(void)
{
	register int i;		/* counter in a for loop */

	/*
	 * generate the table
	 */
	for(i = 0; i < 10; ++i)
		printf("%3d %6d %6d\n", i, power(2, i), power(-3, i));

	/*
	 * bye!
	 */
	exit(0);
}
```

Program #8b: Simple Exponentiation Function

This is the function call, from the same program and (as I wrote it) in the same file. Interestingly enough, this program will still work if the two routines (*main* and *pow*) are in different files!

```
/*
 * compute a power
 *
 * arguments:      int base      what you want to raise
 *                 int n         non-negative integral power
 *                               to raise base to
 *
 * returns: base ^ n (base ** n to you FORTRANers!)
 *
 * exceptions:     none handled; overflow may occur, but there
 *                 will be no indication of it
 */
int power(int base, int n)
{
      register int i;            /* counter in a for loop */
      register int p;            /* resulting power */

      /*
       * do it the obvious, multiplicative, way
       */
      for(i = 1, p = 1; i <= n; i++)
            p *= base;
      return(p);
}
```