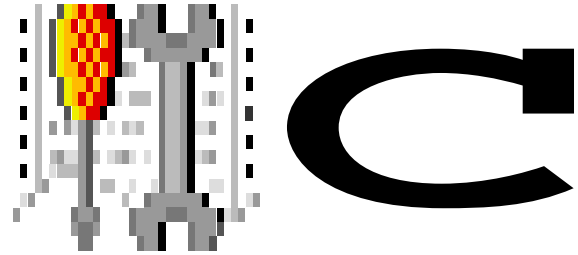


# Frobnicate

FEBRUARY/  
MARCH 1996

Issue 7. The techie magazine for Acorn Enthusiasts.

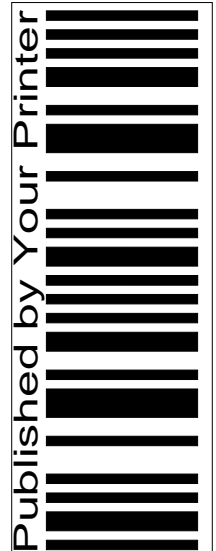


This issue created totally by Richard Murray.



## In this issue:

- 👉 WIMP programming
- 👉 Starting a BBS
- 👉 What's Acorn up to now?
- 👉 Hacking
- 👉 And ***much*** more besides!



## Index:

Page 2 . . . .	Index.
Page 3 . . . .	Editors Page(s).
Page 5 . . . .	Starting a BBS, part 2.
Page 8 . . . .	The Hacker's Diary.
Page 12 . . . .	BASIC from C.
Page 16 . . . .	EyeBall.
Page 18 . . . .	The Hacker's Handbook.

## Credits:

Editor . . . . . Richard Murray.  
Contributors . . Richard Murray, Keith Hall.  
Others . . . . . Hugo Cornwall, Cuisinart Blade.  
Graphics . . . . . Richard Murray.

You may print and/or distribute this document provided it is *unaltered*.

The editor can be contacted by FidoNet netmail as "Richard Murray" at 2:254/86.1 or 'rmurray@arcticbb.demon.co.uk'.  
Feel free to comment or send submissions.

Back issues, stylesheets, notes, logos and omitted articles are available from Encina BBS – netmail editor if you are interested.

The contents of this magazine are © Richard Murray for legal reasons. Full Full credit is given to the individual authors of each article. All copyrights and/or trademarks used are acknowledged.

All opinions stated are those of the article author and do not necessarily represent the opinions of Frobnicate, BudgieSoft or Richard Murray.

All reasonable care is taken in the production of this magazine, but we will not be legally liable for errors, or any loss arising from those errors. As this magazine is of a technical nature, don't do anything you are unsure of. Reliance is placed in the contents of this magazine at the readers' own risk.

Frobnicate is now *managed* by "**Hissing Spinach**",  
the publishing division of BudgieSoft UK.

All contact details et cetera remain the same.

# Editor's Page(s)

Well, the telephone bill arrived, delivered by fork-lift. Good god - it was £500... Half a grand. An Aleph1 PC podule with some RAM. Three modems. A new monitor. Quarter of a RiscPC. Four copies of OvationPro. Excuse me while I slit my throat and wonder why I don't own BT shares, or wonder why I didn't defect to Mercury or CableTel. In fact, wonder why I have, to date, spent nearly £2000 on telephone bills and have so little to show for it.

Comms is an expensive hobby if you approach it as a 'user' and new a 'SysOp'. I tried both, but due to environmental reasons I was a 'user' most of the time. My computer is noisy both audially with the whirring drives and electrically with the lines on the TV screen. As a user I can switch on and off as I please without worrying about somebody phoning.

Back as little as 1994, I was one of the 'better' SysOps. Not meaning I cared more or had more files or anything, but that I owned a fast computer (A5000) with a fast modem (14400bps) and I had plenty of storage space (1Gb). Now, in such a short time, the march of 'doze has meant that every major SysOp has a RiscPC with several Gbs of storage, DAT, CD-ROMs and over 28800bps transfer speeds.

That's the march of technology. Those that bought a RiscPC have to face the fact that the VIDC2 is outdated and the ARM is slow... Compared to the new machines.

One could reasonably wonder what "specification" is actually needed in order to run a BBS. The most important thing is the modem speed. Typically the modern average is 28800bps. But does it take a 40MHz ARM800 to drive it? Can't an ARM3 do the job? After all, the processor can blit data around those RAM pages a heck of a lot faster than the modem can shove it down the telephone line.

One can only wonder. As processors get faster, programs get bigger and slower.

This has been an interesting week for another reason too... After hearing Ian Hislop wittering on (again) about private eye and his inability to say a certain person's name or any anagram of it (except "There's a big fat b\*\*\*\*\* that looks like he's drowning"), I finally went out and bought Private Eye.

Here are some clippings:

**TRICK question: Which foul Tory government decided to dose the tips of artillery shells with uranium, Maggie O'Kane's Channel Four program on 3 January having revealed that such shells were widely used in the 1991 gulf war?**

Answer: The last Labour Government.

Tony Benn's diaries for 22 January 1979, record: "Fred Mulley (defence minister) had proposed that depleted uranium from American nuclear reactors be used for ammunition against Soviet tanks..." He also proposed that "we develop it ourselves and test-fire it".

Benn opposed it but the then prime minister James Callaghan said it must be done "to reassure the Germans". So everyone in the Labour cabinet bar Benn nodded it through.

[Private Eye issue #889, page 10]

If anybody fears that there may be some hidden political agenda here by picking an article that slags off Labour - you're right.

Bunch of ..... I'll stop there as my personal political views may offend some of my more "restrained" readers. :-)

Another bit is where they'll offer you a tenner if you can spot hidden Sky advertising in a Murdoch newspaper [issue 889, page 11].

So, it may be a bit overwhelming for some readers, but if you're looking for a bit of political fun – you could check out Private Eye. In our WHS, it was hiding in amongst the "weird" category – you know, those cool magazines like Fortean Times and Encounters.

For Christmas, mum got me a The Prisoner video. Nice. Did you know that even though the The Prisoner uses 60's technology and has a credit card in which units are removed with a card-punch, it was in fact depicting the modern-day Credit Card several years before they were introduced.

My favourite episode of The Prisoner is the very last. Sure enough the story there is a bit naff as some executive twat decided to wrap it all up so they had to come up with a rush ending. Nonetheless, it takes some kind of genius to create a scene with people running around shooting each other with machine guns, and all you can really hear is "All you need is love" coming from the Jukebox. I watch way too much TV and I've seen many impressive scenes from many impressive films and programs. For technical merit, Jurassic Park was excellent. But this one scene from The Prisoner must surely rank as my all time favourite most powerful scene. On to favourites I suppose. Hmm... One of my current favourite series is the X-Files, followed very closely by "My so-called Life". For anybody that reads my code, the

"Angela Chase" references pertain to that programme. Next up in third place would be Friends, but C4 is showing repeats at the moment, so they're fourth with the latest arrival "Cybill" jumping in at three. Okay, so "Cybill" may be a bit unrealistic and tame (re: The Late Show, BBC2 11/01/96) but sometimes I just want to suspend reality and have a good laugh. Hmm... Maybe it's better if I find reality first.

Adverts. Who can talk about "the best of" and not pick out the best adverts?

In third place is the Pentium advert. Electronic excellence at it's best. So excellently produced that it apparently was rendered on a Mac because the Pentium wasn't up to it.

In second place is the Coca-Cola advert. It's suave, refined and basically a visual and audio treat that doesn't result to naff stuff to push their product. It didn't sway me. I still drink Lime Citrus Spring, but the advert was good.

In first place was an advert I saw only once. The Holsten Pils anti drink-driving advert. I hope the message got through. It was a little "musical" piece, sung by that blonde bloke that hates the 60s and the Internet. Suffice to say, the chorus line went "he's a bas.....!".

Then, I suppose, I'd better run down the worst five things of Christmas, in reverse order:

5. Anything on BBC radio (esp. R4).
4. Emmerdale/Eastenders/The 'street.
3. The Queens speech (Richard - you philistine!).
2. Lilly Savage.
1. Whatshisname Flowers Pops.

Though to win back some "brownie points", I'll say that I did watch (and enjoy!) an old movie called "The Sound Of Music". They don't make 'em like that any more.....

**BYE! HAPPY 1996 Y'ALL!**

# STARTING A BBS 2

Last month I told you to think up a plan for your BBS, and detailed some of the available servers.

So, as promised, I'll show you how to begin assembling your BBS.

The first thing you should do is to back-up the example menus, then remove them from the BBS structure. Instead of editing the default stuff, it's much better to create your own, for several reasons:

1. It teaches you more about how the BBS works.
2. Your "creativity" isn't restricted by what Keith Hall or Hugo Fiennes put together.
3. Don't be so lazy. BBSing is no challenge for the lazy...

For now, we shall examine how to build up a BBS with ArmBBS, as you can get a copy easily by nattering with Keith Hall.

The first thing is to work through the system converting the files. I got rid of the entire MENU directory, but copying it to !Scrap (for temporary reference) and creating a blank MENU directory.

## SYSTEM.PRELOGON

This is the text chucked to the user just before the log-on prompt. Your "intro" screen. This should not contain any ANSI, though I must admit my one contains bold/standard VT100/ANSI codes inserted directly. Edit this to your requirements.

The next thing to be altered was the BBS.LOGIN file to say what is required.

Next month, we'll look at building our main menu. Until then, here is a list of commands available in ArmBBS:

### *Script file commands* [by Keith Hall]

The script file commands are enclosed between the delimiters '{' and '}' and can be separated by either a comma or a newline.

e.g.  
{fg 7,bg 4}

or  
{fg 7  
bg 4}

You can put spaces between the commands but remember to include the ',' or newline.

e.g.  
{  
fg 7  
bg 4  
}

### *Comments in script files*

Comments are permitted in script files, and are denoted by the % symbol. When a single % is placed in a script, the rest of the line is ignored, similar to REM in basic.

e.g.

<p>% This line will be totally ignored % so will this one...</p>	fgbg <no> <no>	<p>0 and 7 changes both foreground and background colours to specified colour no.</p>
<p>{fg 2} %the foreground colour will change, but this text will not be output.</p>	anykey	<p>displays 'Press any key to continue' and waits for a keypress</p>
<p>If you place a second % somewhere after the first, but before the end of the line, ArmBBS will ignore the text inbetween the two %'s.</p>	user	<p>displays the currently logged on user's full name</p>
e.g.	fuser luser calls	<p>" first name " last name " number of calls made so far</p>
<p>{fg 2}{lastlogon} %shows the user's logon time% {fg 3}</p>	timeleft maxtime	<p>" time left for this logon " maximum time for one day</p>
<p>N.B. to get a % sign to be output during script file execution, either use the outchr command e.g. outchr 37 (37 is ASCII for '%'), or place a double % sign....</p>	level lastlogon time date type <filename>	<p>" userlevel " time of last logon displays the current time displays the current date displays the file &lt;filename&gt; evaluating script commands</p>
e.g.	logoff <type>	<p>logs the user off, types are as follows : 0 : immediate logoff, no warning</p>
<p>{fg 3}There has been a 40%% increase in files since last year{fg 2}</p>	logoff <type>	<p>1 : immediate logoff, displays script file 'Logoff' 2 : prompts user for logoff, displays script file 'Logoff'</p>
or...	logoff <type>	<p>2 : prompts user for logoff, displays script file 'Logoff'</p>
<p>{fg 3}There has been a 40{outchr 37} increase in files since last year{fg 2}</p>	logoff <type>	<p>2 : prompts user for logoff, displays script file 'Logoff'</p>
<p><i>Descriptions of commands</i></p>		
Command	Description	listusers
cls	clears the screen	menu <filename>
fg <no>	changes text	<filename> and waits for a valid keypress
	foreground colour to <no> between 0 and 7	outchr <ascii chr>
bg <no>	changes text background colour to <no> between	outputs the specified chr number as an ascii chr.  How menus are defined

```
{
defitem <key>    <key> is the required
                 key for this option
display <chrs>  <chrs> is a string
                 displayed when this
                 option is selected
access <level>  <level> is the minimum
                 userlevel allowed to
                 select
[
                 denotes start of
                 command list for this
                 option
.
.               place command(s) for
.               this option here
.
]               denotes end of command
                 list
enditem         denotes end of menu
                 option definition
}
```

Coming next month, an introduction to ArcBBS commands.

Then, once we have the commands and major differences for both systems available as reference, we can begin to build up our BBS menu structure.

e.g.

```
{
  defitem L
  display List users
  access 20
  [
    listusers
    anykey
  ]
  enditem
}
```

This defines a menu option activated by pressing 'L' which then displays "List users" if the current user has a userlevel of 20 or greater. The commands list the users and then asks for any key to be pressed.

N.B. All acceptable script commands can be used in menus, including 'type' and 'menu'.

**Only one has written all these...**

!Door_Fix	Auremia	AR_Info
CastAVote	Cdecdec	DoorDocs
DPAutil	FunBASIC	HappyHak
Help	InfoSys	LastUsers
Linker	MOS	ONLINE
Parlez	ReadTasks	ReadUser
RecordAT7	RecordUsr	Security
SysOpChat	TopFeatur	TTXDoor
UserEdit	ViewFile	!VoteEdit
VoteConv	VoteModule	

**And converted all these...**

!AcroEng	!Breakout	!SpeedyT
!Tetris		

And have *even more* updates and *new* programs to come...

...and has done *all* of this **FOR FREE.**

***...that'll be the BudgieSoft. :-)***

BudgieSoft would like to make it clear to all readers that the demonstration release of **!CAV3MOD** is *NOT* compatible with the newer releases of **VoteModule.**

For the adventurous, basically the old VoteModule counted vote 1 as '0' and vote 2 as '1'. Now vote 1 is '1' and vote 2 is '2'. Look for "VoteModule\_GetVote" and "VoteModule\_GetVoteHeader".

# The Hacker's Diary

This file was downloaded by me several years ago, and is reprinted here because, basically, I think it is interesting...

Now, as if often the case with these "Hacker" stories, it's quite likely to be a fabrication – especially that part about pressing CTRL and SHIFT. Yeah... Quite... I'm sorry readers, I don't buy a word of this. But hey – it's a good read. :-)

I've edited out anything that could be a bit dubious, like "so-called" passwords and company names. Yes, I've added comments, in this type of text (Homerton) so you can't miss them.

## Diary of a Hacker

### Part 1

What I am about to write about happened in October of 1988, right after I had turned 16.

Yes, I had a Computer.

Yes, I was a hacker.

Yes, this is my story.

This story is all true. The events in the story are as real as the noses on your faces. Any attempt to change this story would be boring, and unnecessary.

Hello. My Handle is The Cuisinart Blade, (I have since changed it) most people called me C.B., just as another nickname. I got into computers because I always had a knack for games, programs, etc... I loved the feel of the keyboard as I typed, especially around midnight, when school was only 7 hours away. That \*\*\*\*head in your 1st hour waiting to pick on you at the stroke of the clock. What little that he knew. I loved the feeling of flying through the 'net, talking to the other hackers like me, always trying to impress one another. They came in all fashions of handles: The Ax Murderer, psycho, The Hacker Kid, Phobia, etc...the list is nearly endless. And each with an utterly unique personality, so far advanced from the morons we all remember at high school. I loved talking to these "Wanderers of the Wunderland" so much, I would call all over the United States just to do that. I remember back in 1988, 9600 was almost as uncommon as the newest modems of today. I loved to suck down that Jolt [Soft drink? Beer?–Ed], turn on the overhead light in the basement, and hack until morning. Then go to school to have that enormous football jock try to scare me into submission. I will admit that, yes, he did scare me. But what a fool he was, thinking that pure brawn was power. *Knowledge IS Power*. I never did good in High School. Knowledge is not there, it was in the 'net! THAT was where I was a King. That was where I was a God. ANYONE who has truly roamed the 'net or been in that type of world could tell you that, hands down. I loved running home, or taking the bus, right into my basement!

Homework? Bah! What a waste of time it was, compared to the thrill of hacking? Homework was little more than review. I sat down and got onto my computer and began to type. I called a rather large BBS in which, hackers frequented. Nowadays they are considered "Wandering" Boards. They appear to wander because they change phone numbers all the time. Anyway, I called up my favourite pirate BBS and went onto multi-nodal chat to talk to the others about their exploits. Some of them had told me of their abilities to smash entire phone networks, others had told me of their A-typical grade changes, (yawn). But most of all I loved to brag. The unwritten rule in being a hacker, you're nothing unless you have something to brag about [But the best "hackers" keep their mouths shut–Ed]. At the time, I didn't, so I decided to get a name in the computer Hacking Club and do just that: Hack.

At first I wondered where I should go, what I should do. I had no idea where to begin. I had been all over the world in the 'net, and nothing to show for it, other than just knowing of everyone, or everything. I needed more. That is the answer to everything in the mind of a hacker, the *NEED* to know. I decided to try something small.



That night, my friend, (who I will call Dr. Antristo) and I, called up another pirate BBS and downloaded the number of a rather prominent business, (at the time) and decided to call. Our biggest fear was being traced, so we got together and built what is nowadays almost routine, (from what I hear) a BOX.

I made the call and found myself looking into a Menu System, that asked for a NAME, and Security ID. I had no idea about any security ID, or anyone who worked there, so I decided to hang up and try in a day or two, after I did some "Checking Up" on their Company.

Wednesday came and went, I checked into the directory, and looked for a name, someone in the company. Personal passwords were not being used at this company, so figuring that out was not important, but finding out the *MAIN USER SYSTEM PASSWORD* was.

No problems.

After finding out about a fellow who worked there, I called again. This time the menu popped up again, and I typed his name again it asked me for a system password. I knew the voice number to the company, so I dialled it on my regular phone line and got some reception lady, you know the type, not too bright, and just a tad naive. I told her I was in the Data Processing Department and was new, and had forgotten the system password. It took her all of about 10 seconds to tell it to me. She didn't even bother for proof. So with that in my head I went ahead and typed it.

I was in. My buddy watched, as I began to move through the menus, I got to check out the general bulletin listing, several up-and-coming projects, (heh heh 486?) and even got to see who was late to work this morning. I wanted to get some specs on the new things they had in-store for the commercial market, but I was unsure where to download them to. I was fearful of doing that to my own house, so I figured I would find another place. After rummaging through the place for about 15 minutes, I left.

Personally, I was ecstatic. I had done it. I had gone into a rather large business and gotten to look through their system. But I needed proof, or so I thought. I wanted to show the Hackers of America, I was as good as them. I wanted what we all wanted. To be noticed and respected by the people who I would call my peers. Now I will admit that there were *MANY* who were far better than me. There were guys who didn't pay a penny for any phone calls that they made. Ma Bell was sure p\*\*\*ed! There were others who were what I and others referred to as Professional Pirates. These were the guys who not only wrote their own software, but cracked anything they came across, knew where to get anything, and just seemed to be great at anything they wanted to be. These were the guys everyone tried to be like, they were the football stars, homecoming kings, and all-American heroes of the 'net [Barfacious-Ed].

To us they were as close to god as the pope. Turns out that they were no older than most, but they were smarter, or had a head start. These guys were, and many of them still are, unstoppable. I got talking to another pirate later that evening, after I had gone into the company mainframe, and told him of what I had done. I thought at first he'd laugh at me, but instead, he practically begged me for that password. He said everyone in this system at that moment would give me any Wares [Wares/warez means, basically, pirated software - Ed] I wanted. Within 2 minutes of telling this guy, the multi-node chat was filled with over 20 different hackers, pirates, and onlookers asking for the password, offering me wares, and other numbers in exchange. One guy, I will call Harlock, offered me something none of the others could. The ability to USE TelCo [TELEphone Company-Ed] all I wanted for free. He said he would teach me how to use their LOOP Numbers and not get caught, also teach me to call other places, drop to their DOS shells [TelCo running DOS???-Ed], and make hidden directories where I could put my "borrowed" wares.

All for a password.

I accepted.

Without hesitation, he gave me a number to his "Wandering" BBS. I disconnected with this one, and logged onto his. It was a long distance call in the 216 area code. I didn't want to stay too long, otherwise my parents would get REAL mad. I arrived at the logon screen to be greeted with an enormous ANSI of a kings crown. Followed by a system password. The SysOp of the board typed it in for me, and took me directly into chat. He told me that TelCo has what are called LOOP Numbers, numbers that are out of circulation, but can still be used. Billing is not used, because the company writes it off as an expense on their annual tax bracket as for Public Relations or Advertising. They don't check up on them, because with over 500 million nodes, and lines in this country they don't bother with the "lost" ones. (Currently there are around 75,000 LOOP Numbers in the 312/708 area [Hmmm...-Ed]). He told me that in order to use one, you must find one, by what he called garbage screening [Better known as "Trashing"-Ed]. He

told me that around where he lived they just chucked all the paper in the garbage bin and left it for the trash men on Friday (he thanked god that recycling hadn't REALLY started yet). He advised me to do the same. Once I had the numbers, keep a blacklist of them, (of which only a quarter still work to this day).

Then he explained how to do it. He said I was to dial the number, wait for 2 "Chimes" then hit the Shift and Ctrl buttons [I smell a little pile of bull around about here—Ed]. With Modem, (I still don't know how, but it worked so I am telling you) it would give me a dial tone, and wala! [Obviously didn't do good at French, "Voilà"?—Ed]

I could then proceed to call anywhere in the world, under TelCo's tab. Then he asked me for the password which I happily gave him, (I may be a white-collar crook, but I am an honourable one). After writing it down, (I guess) he told me I was welcome to look around his system. I thanked him for the info, then logged offline.

I was info hungry! I had written the directions for the LOOP numbers down and wanted to see what I could do. I told Dr.Antristo to power up the MouseMobile (his Jalopy) 'cause we were gonna go shopping this evening.

Around midnight, Dr.Antristo and I got to Oakbrook, a rather large communications hub for the Greater Chicagoland area. TelCo had just thrown its garbage away and there was no one around. I walked over to the bin, with doc' in the Getaway Mouse, and looked for any kind of security measure. It had a padlock, that was no where to be found. I pushed the door open and took a look. Bags of paper and more bags. From what Harlock had told me, I had hit the jackpot. I climbed in, (you must understand, that this was the cleanest garbage I had ever seen) and began to look for anything that had to do with numbers. I also grabbed a pair of hefty bags full of paper with what looked like numbers then got to the car and we returned home. I dumped all the stuff onto the floor and began to rummage through it. I found a lot of office memos to call some Steve guy, and letters of recommendation, as well as a few resumes'. Then luck. A list of numbers that were said to be disconnected or transferred to station (52?) for standard procedure. Another list detailing numbers offline or as it put it, "Out Of Circulation". At last, O.O.C.'s! I wanted to know, I had to know. I put one of them into my Telix directory and ran it through. After a few seconds of prayer, I heard a high pitched noise, almost like a modem, but it was off a few bars. Then I heard what I can only describe as a "chime", then another...I hit the Shift and Ctrl keys, and what almost sounded like line shifting, I got a dial tone!

Success!

Dr. Antristo and I ran about the room hollering and yelling our thanks to the great Jolt god and we sat down at the computer and typed in the name of a long distance BBS, we could never call. Again, success. It didn't connect because it was busy, but it went back to dial tone again, and we tried another. The BBS was called Crystal Palace, and it was somewhere in Canada. We logged into it, ran through the usual new-user crap and spent an hour talking to the SysOp, who was watching us at the time [as one does when you talk to one—Ed]. We decided not to try again, until the phone bill came for the month. One week later it arrived. I tore it apart to find all long distance calls done. Only one to my Aunt Kathy in Michigan! The rest were local! I was so happy I didn't care if my father was yelling at me about opening his mail. I called Dr. Antristo and told him the great news. With this, we could do some real neat things, like call all those other Hacker Boards we could never reach.

We decided to call the most interesting on your list, one called ToC: The Obsidian Crackhouse. It was in Germany, a place where there seemed to be a ton of Elite Pirates. We dialled into the LOOP Number and then made the overseas call Connect 9600bps!

We tied into the BBS to find that we couldn't access, it was a private BBS. So we settled for something in our country, a place in the 404 Area Code. A BBS called The Nutcracker Elite. We got in and "registered" No, we weren't cops, No, not Federal Agents, etc.. then we looked around to see what this guy had. Files I had never seen before, stuff from England, California, Germany, Holland, Ireland, etc... Lists so long I couldn't believe what I was looking at. I wanted to download everything I saw [I know the feeling—Ed], but I was still afraid of what might happen, trying to do that. So I figured I would give ol' Harlock a call. I finally got hold of him on a Friday night, around one in the morning, and asked him what to do. He told me what he did. He would use the LOOP Number, then call another company, drop into their DOS, create a hidden directory, then call out again, under THAT company's number. It boggled my mind, so I had him explain it again. The LOOP Number worked for you in several ways. If you use a LOOP Number to call out, if you call from another HUB, it would look like you were calling from that area. So anyone tracing would track it to that location, and if they were good, they would only get as far as the LOOP Number origin, in which case you use another number. He warned me that the BOX was the only

thing keeping TelCo from running a trace from their “hidden” line to my house. So I continued to use it.

That night I called some bank in the 804 Area Code. Found their general logon password, (from a Hacker in Vermont) and proceeded to drop to DOS. I looked at their inferior setup, all memory and nothing to show for it. I created a hidden directory, then exited and called out from that location using one of their company lines to another BBS in the 213 area code, a BBS named Psychotic Interlude. I got online and downloaded as much as I could. (a whopping 3 files).

After disconnection, I did another download to my house through the LOOP Number. It worked. The entire evening I was online at one place or another. I had gotten about 15 meg worth of wares from around the country [If it takes my 14400bps about 12 minutes to transfer a Mb, this guy must have been on for absolutely ages at 9600bps–Ed]. This Bank had so much memory it never would have missed it.

I was really beginning to enjoy this. The following evening, I went online at my usual Hacker hangout and chatted with a few guys, the topic of the day was someone in Georgia was saying that he could take down the entire phone grid in that state. And after doing what I did, I believed him.

October 24th, 1988

I went to Dr. Antristos house to pick up the software of what we hoped to be our new BBS. I wanted to be a SysOp. too bad that he didn't have it. After discussing what we could do with our abilities, we wanted to see if we could get into a large corporation. Watch out world, we were arriving!

That night, after slurping down a 24 pack of Jolt, we listened to some Black Sabbath and Led Zeppelin, (my favourite song by them is Cashmir on Physical Grafitti Album).

We entered the world of the hacker. We went to our usual LOOP Number and then dialled into a corporation in Chicago [Big Blue–Ed]. Well, we got to their commercial menu, where they tell you all about their products, (Yawn) and how they're the best. If they were smart they wouldn't have agreed to giving royalties to the creator of their logon program for all IBM-compatibles, and regulars. Anyways, we managed to snoop into general information, but Dr. Antristo and I wanted more. We wanted much more. So I decided to try that newbie crap with the reception lady again. I figured that if it worked once, it would work again.

The lady, who turned out to be a male night security guard, scolded me for calling so late, and muttered something about yuppies and their late hour workings. He looked around on the desk of wherever he was and gave me a password to enter the mainframe, (I still laugh at this chucklehead they had hired) and in less than 20 seconds I was looking at commands for the office. Turns out he gave me a Password that is not for general use. I got an office password. I was looking at a menu that asked me if I wanted to shut off the lights! I gave out a manical laugh! I was god, I was immortal, I was....hacking. I decided not to turn off the lights, but I wanted to look at future projects slated for the next year. After browsing around, I got bored and looked into the security matrix, and found a universal password and name collection. I was feeling a bit evil that evening so, I switched, added, removed, and changed some passwords...heh heh. Cruel to the people who think they're tough shit! I left the company with a big grin, and a lot of experience. Dr. Antristo told me that we ought to go after other things. I wanted to know so much about everything, so I had it in my head I was going to do just that.

Only thing was, how to do it!

And coming in the next issue of Frobnicate is part two of this exciting (and half believable (but only half!)) story!

There are three luscious (?;?) parts in all... Don't miss them!

# C *FROM* BASIC

Now I'm fully aware of the problems some of you have with this. You know a reasonable amount of BASIC, but you'd like to learn to program in C for whatever reason, be it the portability aspect or just because compiling makes it much harder for opportunist "thieves" to rip your precious code.

You got yourself a C package. Here I'll assume it's Acorn (Norcroft) C release 4, but there are alternatives such as Beebug's EasyC/EasyC++ or Acorn C/C++ release 5 or (for the die-hards) 'gcc'. It doesn't really matter. The initial language matters are the same.

There seems to be some unwritten rule that a C coders first example program plonks up "Hello world!" on the screen. Well, I've never been exactly normal or totally sociable, so we can forget about that right away... :-)

```
10REM >DemoProg1
20INPUT "What is your name ";name$
30PRINT "Your name is..." + name$
40END
```

Okay, so it's a pathetically useless program, but it is the sort of thing you find in "Learn a programming language" type books – so we'll use it.

We encounter five problems here...

1. How are C programs laid out?
2. How do we make comments in C code?
3. How do we input something?
4. How do we print some text?
5. How do we "END" a compiled program?

Okay...

1. For most C programs, you must ALWAYS have a function called "main". The exception is code that isn't run (like a module), but you shouldn't worry about that. We won't be dealing with that for ages.

The correct prototype for a basic C program is (you can move declarations to .H files):

```
Initial comments (version/name etc).
#include.
Declare variables.
Declare functions (depending on code style).
```

```
int main(void)
{
    Commands
}
```

Firstly the comments, refer to part 2 (below) for details. The initial comments should at least say what the program is and what version it is.

Secondly, #includes. You must "include" certain library code in your program. What

you include depends largely upon what you are trying to do, but at the very least you should include a library called "stdio.h". For 'system' libraries, you include them like:

```
#include <library.h>
```

And 'user' or 'OS dependant' libraries are included like:

```
#include "library.h"
```

This is one of the many eccentricities of C. Get used to it. Things get weirder.

Next, variable declarations. Unlike BASIC, you cannot just 'use' a variable when you need it. ALL variables must be defined. Also unlike BASIC, variables come in many different flavours and even some cases where the variable is there...but isn't. :-)

Next are the function prototypes. There are two schools of thought here. One is the say that every function must appear 'above' any function that calls it, thus placing "main" at the bottom. This is like conventional PASCAL. The other school of thought (and one that I subscribe to) is to put "main" first. Below that come procedures used in "main", then other procedures until all the fiddly bits are at the bottom. Then a rough cut'n'paste to group code into sections like "file handling bits". This often requires the use of function prototyping (defining a 'rough copy' of the function call/return parameters before it is used.

You have no need to learn about this yet, but a simple BASIC example will illustrate:

```
REM >!RunImage
DEFNmyfunction(a%):=b%
PRINT myfunction(12)
END
:
DEFNmyfunction(a%)
```

```
b%=a%*2
=b%
```

Basically, it tells the compiler how to handle the function if it has never seen it before. Basic will flip out at this though. :-) Enough already...

```
int main(void)
```

Now many C books will show this to you as "main()". This is correct, but the proper form is "int main(void)". It takes no parameters (usually, we'll cover this later) and returns an integer.

```
{
Code segments are enclosed in curly
brackets. Then they are treated as one block
of code. Curly braces follow
procedure/function headers and hold the
contents of multi-line IFs or so forth.
```

Commands

Here you put the body of your program. It can run on its own or call other functions.

```
}
For every opening curly brace, there is a
closing curly brace.
```

2. Comments are denoted by the sequence `/*`. You can have as much comment as you like, though you cannot nest comments. Once your comment is done, use the `*/` sequence to end the comment. In BASIC, this is like putting a REM at one and and a MER at the other. However unlike BASIC, your comment can span many lines, even entire chunks of code, as easily as it can span one character. A comment can, interestingly, be stuck right in the middle of your code. Consider this:

```
E /*energy*/ = m /*mass*/ * c2 /*speed of light2*/
```

Now nobody in their right mind should suggest that you write code like that, but just consider the flexibility of it in debugging. What does BASIC offer? A “REM” which, when found, will cause BASIC to ignore the rest of that line. Ho hum.

Certain versions of PASCAL have two types of comment bracket { } and ( \* \*) allowing nested comments. C does not, and you have to watch for this when debugging. The use of !Zap is recommended, as all C comments appear in green.

3. Unlike BASIC, there is no generic “INPUT” statement. There are two ways to input information from the keyboard. The first is “scanf” which is fiddly and not really suited for our purposes. The second is “gets” which means ‘get string’ and is exactly what we want.

Other ways are to treat the keyboard as a device and try to perform file reading operations on it. C allows you to do all sorts of totally nutty things!

4. The simplest way is to use “printf”. That means ‘print formatted’. Get immediately out of the habit of thinking like BASIC.

```
printf("User "+username+" on line "+STR$(port));
```

won't work.

Instead you want something like:

```
printf("User %s on line %d\n",username,port);
```

I'll admit, it seems quite weird. I know, I've been there. But believe me, once you get the hang of it, you'll prefer it to BASIC's way.

The “%s” means ‘substitute the specified string’ (first parameter – “username”) and the “%d” means ‘substitute the specified integer’ (second parameter – “port”). The “\n” is the code that represents ‘newline’. This is just an

example, and the conversion specifiers, as they are known, will be explained when we come to use printf in more imaginative ways.

5. Finally... Ending... You simply call the “exit” function, passing across a number or 0 to indicate if the program was successful or terminated due to error. In Acorn C, I use the defined macros “EXIT\_SUCCESS” and “EXIT\_FAILURE”. See how your compiler handles the “exit” function.

So now we have done that, we'll build up our program...

```
/* Demonstration program 1
   By Richard Murray
   For Froblicate #7, February 1st 1996
*/

#include <stdio.h>

char input[81]; /* Space for 80 characters */

int main(void)
{
    puts("What is your name ?");
    gets(input);
    printf("Your name is...%s\n");
    exit(EXIT_SUCCESS);
}
```

Right, the “exit” isn't strictly necessary. The next thing you probably noticed is the “puts” function. Now if I wanted I could have used printf. However if there is no conversion necessary (puts can't do “%s” or “\n”), then again, quicker code is generated using puts.

Next you'd have noticed that where it says room for 80 characters, I have in fact reserved room for 81. C uses a ‘null’ character to indicate the end of a string

because C doesn't understand strings. It understands a pointer to a bunch of bytes that represent a string. Phew! Tough work this C stuff isn't it? :-)

Surely from your BASIC programming you'd come across SWIs that return strings that are not terminated with carriage returns. Therefore you have to write a special routine to step through an array extracting the string until you come to a '0' byte??? Almost every WIMP programmer has written this little routine.

Next, yes more, is this matter of the semicolon. Like PASCAL, C lines are usually terminated with a ";". The rules are eccentric and something along the lines of:

All lines must be terminated by semicolons except lines followed by a "{", for example:

```
if (something == TRUE)
{
    printf("It's true!!!\n");
}
```

But in such a case, the curly brackets are unnecessary and you could write:

```
if (something == TRUE)
    printf("It's true!!!\n");
```

But C will allow you to abbreviate even further until you come to:

```
if (something == TRUE) printf("It's true!!!\n");
```

And now we are back to our single line.

Note that function prototypes end in a ";":

```
int myfunction(int a);
```

But actual function code does not:

```
int myfunction(int a)
{
    a++;
    return a;
}
```

If you are currently wondering about comments requiring semicolons after, now is time to take a very long tea break... No, they don't... :-)

Incidentally, C is pretty "free form". You can make use of white-space (unlike BASIC) as the compiled C program doesn't rely upon the formatting of the source code.

Our program could just as easily look like:

```
/* Demol by Richard Murray, Frob. #7, 1-Feb-96 */
    printf("\n%d plus %d is %d", a, b, c);
}
#include <stdio.h>
```

```
char input[81];int main(void){
puts("What is your name ?");gets(input);
printf("Your name is...%s\n");exit(EXIT_SUCCESS);}
```

And it'd produce the same output. So be neat, be tidy. Forget the BASIC ethos of "keep it short keep it small". Instead make your programs readable. Include comments where necessary and give your variables and functions useful names, like "input" and "calculate\_blood\_cholesterol\_level" instead of "a" and "b". Okay, the second example was a bit OTT but it proves the point.

Leave obfuscation to the yearly contest.

Until the next issue, here is a little something for you to examine. It introduces some concepts we'll look at next time...

```
#include <stdio.h>
int a,b,c;int addition(int x,int y);
int main(void){
    printf("Enter the first number : ");
    scanf("%d",&a);
    printf("Enter the second number: ");
    scanf("%d",&b); c = addition(a,b);
    printf("\n%d plus %d is %d",a,b,c);
}
int addition(int x,int y){
    return (x+y);
}/* Sorry about the squeeze, had to stuff it in!*/
```

# EYEBALL

Right, this new article will take an oblique look at technological stuff around.

The first thing on the agenda is your *real* flexible friend, your SWITCH card. Much easier than credit cards and doesn't cost as much. Friends have told me that foreign cards aren't always recognised in the UK. However if you look on the back of your SWITCH card, you'll see it says "CIRRUS". So plop your SWITCH into a French cash dispenser and gasp as you get English instructions, French cash and a cute little ticket (tee-kay)...



Which is quite funny when you consider that some UK bank cards won't talk to other UK bank cards.

Next up, the worlds biggest network, and getting so big they'll soon run out of IP address allocations (despite the massive unused chunks that were

given out by ARPA in the early days). Now it is true that the Internet and these STBs will explode into a messy information system – which can only be good for TelCo. People “surf the net” day in day out getting hooked on the piles and piles of data therein, and only dying when the bill crashes through the floor.

Now we come onto the topic of the day. We come on to the death of our beloved computer system. We come to the Apple<->Acorn merger, which is rather funny considering the advertising policies of the former. I, like numerous people on Comp.Sys.Acorn.\* newsgroups were hit with panic. Time to pull out the nappies and riot gear. But I have thought long and hard about this predicament. Whilst it is all speculation and us mere mortals don't really know the whole story – time will tell.

Look at it like this. Acorn are now “in” with Apple, digital and Oracle. We can forget the “games machines” and the “obsoletes”. So who is more likely to crack the PC domination than the above group? Companies try to break into the PC world and end up making PCs to survive. It looks as if Acorn was heading that way with Acorn badged PCs.

But what now? Well, Acorn is more likely to be a player in the CHRP system. The StrongARM, if it is a worthy hunk of silicon (where is it???)



But what about our OS? You see, CHRP runs MacOS which is, as one poster put it, “fundamentally flawed”. You can’t run ’doze as the main OS as that is only just reaching something nearing RiscOS2 specification.

But this Apple/Acorn/digital mix isn’t as bad as it sounds. Apple are best at good sounds and good graphics. Acorn are best at really decent operating systems. digital are best at “well fast” processors.

After thought, rethought and examination; whilst the future isn’t totally clear, it is beginning to look like Acorn have just pulled off the final wraps of the biggest coup this decade – in the computing fraternity. People predicted the PC setups to crash&burn around the end of ’96 to mid ’97. Nobody was sure *how*. Could this be it? They’ve three months until mid ’96.

I await with interest and amusement. I never thought much of PCs anyway.....

So on to RiscOS. Early reports were that Oracle were using a proprietary OS. Now the idea seems to be that they’ll be using RiscOS. This could, in theory, sling a few zeros to the end of the RiscOS userbase count – and make it a very widely used system indeed.

Whatever happens, I’m not going to buy a RiscPC. The next year is going to be interesting I think. So I’m waiting.

And watching.

The future...is 20,000 lux.  
The future...is Acorn.

## New from *BudgieSoft*...

### Auremia:

Released two hours before this was written, Auremia is a neat little door the will plot a changing ’tag’ on your menus. This is my first door written in C.

### CastAVote 3:

The alpha version has finally been released. You’ll find it on ArcTic and other good BBSs.

### VoteModule:

Is now sufficiently functional to run CastAVote properly. The module has been released. Expect a new !CAV3MOD program to follow.

### AND COMING SOON:

#### CAV PRM:

Yes, the CastAVote programmers reference manual. Following a few comments by Andrew Lobel about my lack of documentation, I am putting together a totally concise guide to CastAVote.  
In Ovation format.

#### Other stuff...

### AND IN TESTING HERE *ONLY*:

#### InfoSys2:

The InfoSys project has been revived. Send a message to 2:254/86.16 with **HELP** in the body of the message...

# *Hacker's* Handbook

Reading now, it seems funny that this book was banned. Ah well... Have a nostalgic peek into the days of 300bps modems with this serialisation of "The Hacker's Handbook". It is interesting to note that the first "hacker" was a girl. :-)

ASCII version compliments Electronic Images - Gizmo

Century Communications present THE HACKER'S HANDBOOK, copyright © Hugo Cornwall.

All rights reserved. First published in Great Britain in 1985 by Century Communications Ltd. Portland House, 12-13 Greek Street, London W1V 5LE. Reprinted 1985 (four times). ISBN 0 7126 0650 5 Printed and bound in Great Britain by Billing & Sons Limited, Worcester.

## INTRODUCTION

The word 'hacker' is used in two different but associated ways: for some, a hacker is merely a computer enthusiast of any kind, who loves working with the beasties for their own sake, as opposed to operating them in order to enrich a company or research project - or to play games.

This book uses the word in a more restricted sense: hacking is a recreational and educational sport. It consists of attempting to make unauthorised entry into computers and to explore what is there. The sport's aims and purposes have been widely misunderstood; most hackers are not interested in perpetrating massive frauds, modifying their personal banking, taxation and employee records, or inducing one world super-power into inadvertently commencing Armageddon in the mistaken belief that another super-power is about to attack it. Every hacker I have ever come across has been quite clear about where the fun lies: it is in developing an understanding of a system and finally producing the skills and tools to defeat it. In the vast majority of cases, the process of 'getting in' is much more satisfying than what is discovered in the protected computer files.

In this respect, the hacker is the direct descendant of the phone phreaks of fifteen years ago. Phone phreaking became interesting as intra-nation and international subscriber trunk dialling was introduced, but when the London-based phreak finally chained his way through to Hawaii, he usually had no one there to speak to except the local weather service or American Express office, to confirm that the desired target had indeed been hit. One of the earliest of the present generation of hackers, Susan Headley, only 17 when she began her exploits in California in 1977, chose as her target the local phone company and, with the information extracted from her hacks, ran all over the telephone network. She 'retired' four years later, when friends started developing schemes to shut down part of the phone system.

There is also a strong affinity with program copy-protection crunchers. Most commercial software for micros is sold in a form to prevent obvious casual copying, say by loading a cassette, cartridge or disk into memory and then executing a 'save' on to a blank cassette or disk. Copy-protection devices vary greatly in their methodology and sophistication and there are those who, without any commercial motive, enjoy nothing so much as defeating them. Every computer buff has met at least one cruncher with a vast store of commercial programs, all of which have somehow had the protection removed - and perhaps the main title subtly altered to show the cruncher's technical skills - but which are then never actually used at all.

Perhaps I should tell you what you can reasonably expect from this handbook. Hacking is an activity like few others: it is semi-legal, seldom encouraged, and in its full extent so vast that no individual or group, short of an organisation like GCHQ or NSA, could hope to grasp a fraction of the possibilities. So this is not

one of those books with titles like Games Programming with the 6502 where, if the book is any good and if you are any good, you will emerge with some mastery of the subject-matter. The aim of this book is merely to give you some grasp of methodology, help you develop the appropriate attitudes and skills, provide essential background and some referencing material – and point you in the right directions for more knowledge. Up to a point, each chapter may be read by itself; I have compiled extensive appendices, containing material which will be of use long after the main body of the text has been absorbed.

It is one of the characteristics of hacking anecdotes, like those relating to espionage exploits, that almost no one closely involved has much stake in the truth; victims want to describe damage as minimal, and perpetrators like to paint themselves as heroes while carefully disguising sources and methods. In addition, journalists who cover such stories are not always sufficiently competent to write accurately, or even to know when they are being hoodwinked. (A note for journalists: any hacker who offers to break into a system on demand is conning you – the most you can expect is a repeat performance for your benefit of what a hacker has previously succeeded in doing. Getting to the ‘front page’ of a service or network need not imply that everything within that service can be accessed. Being able to retrieve confidential information, perhaps credit ratings, does not mean that the hacker would also be able to alter that data. Remember the first rule of good reporting: be sceptical.) So far as possible, I have tried to verify each story that appears in these pages, but hackers work in isolated groups and my sources on some of the important hacks of recent years are more remote than I would have liked. In these cases, my accounts are of events and methods which, in all the circumstances, I believe are true. I welcome notes of correction.

Experienced hackers may identify one or two curious gaps in the range of coverage, or less than full explanations; you can choose any combination of the following explanations without causing me any worry: first, I may be ignorant and incompetent; second, much of the fun of hacking is making your own discoveries and I wouldn’t want to spoil that; third, maybe there are a few areas which are really best left alone.

Nearly all of the material is applicable to readers in all countries; however, the author is British and so are most of his experiences.

The pleasures of hacking are possible at almost any level of computer competence beyond rank beginner and with quite minimal equipment. It is quite difficult to describe the joy of using the world’s cheapest micro, some clever firmware, a home-brew acoustic coupler and find that, courtesy of a friendly remote PDP11/70, you can be playing with Unix, the fashionable multitasking operating system.

The assumptions I have made about you as a reader are that you own a modest personal computer, a modem and some communications software which you know, roughly, how to use. (If you are not confident yet, practise logging on to a few hobbyist bulletin boards.) For more advanced hacking, better equipment helps; but, just as very tasty photographs can be taken with snap-shot cameras, the computer equivalent of a Hasselblad with a trolley-load of accessories is not essential.

Since you may at this point be suspicious that I have vast technical resources at my disposal, let me describe the kit that has been used for most of my network adventures. At the centre is a battered old Apple II+, its lid off most of the time to draw away the heat from the many boards cramming the expansion slots. I use an industry standard dot matrix printer, famous equally for the variety of type fonts possible, and for the paper-handling path, which regularly skews off. I have two large boxes crammed full of software, as I collect comms software in particular like a deranged philatelist, but I use one package almost exclusively. As for modems—well, at this point the set-up does become unconventional; by the phone point are jack sockets for BT 95A, BT 96A, BT 600 and a North American modular jack. I have two acoustic couplers,

devices for plunging telephone handsets into so that the computer can talk down the line, at operating speeds of 300/300 and 75/1200. I also have three heavy, mushroom coloured 'shoe-boxes', representing modem technology of 4 or 5 years ago and operating at various speeds and combinations of duplex/half-duplex. Whereas the acoustic coupler connects my computer to the line by audio, the modem links up at the electrical level and is more accurate and free from error. I have access to other equipment in my work and through friends, but this is what I use most of the time.

Behind me is my other important bit of kit: a filing cabinet. Hacking is not an activity confined to sitting at keyboards and watching screens. All good hackers retain formidable collections of articles, promotional material and documentation; read on, and you will see why.

Finally, to those who would argue that a hacker's handbook must be giving guidance to potential criminals, I have two things to say: First, few people object to the sports of clay-pigeon shooting or archery, although rifles, pistols and crossbows have no 'real' purpose other than to kill things—and hackers have their own code of responsibility, too. Second, real hacking is not as it is shown in the movies and on tv, a situation which the publication of this book may do something to correct. The sport of hacking itself may involve breach of aspects of the law, notably theft of electricity, theft of computer time and unlicensed usage of copyright material; every hacker must decide individually each instance as it arises.

Various people helped me on various aspects of this book; they must all remain unnamed—they know who they are and that they have my thanks.

## WEREWOLF SOFTWARE

announce...

# CDtracker

A comprehensive database program  
for audio CDs with advanced playing  
facilities

Contact us now for more info  
and download a demo from our support BBS!

Werewolf Software • 23 The Spinneys, Bromley, Kent BR1 2NT  
Tel: 44-181-467 1138 • Support BBS: 44-181-289 6003 (6pm-8am)

Do you want to follow up  
Froblicate? Do you want to  
discuss the issues raised?

Now you can...on FidoNet.

Simply ask your host to  
link into **FROBNICATE**, now  
available through the MNL.

### ADVERTISE IN FROBNICATE...

It's simple. Just sent Richard Murray a copy of your advert (don't private upload on ArcTic).

You can submit: ANSIs, AcornDTP, 1stWord+, HTML (not proprietary), Impression/Style, Ovation, Draw (recommended), ArtWorks, WordPerfect 5, AMIPro and plain old ASCII.

IT IS **FREE**.

File uploads to : Arcade +44 181 654 2212, private file to "Richard Murray".

UUcodes (under 10K total) to: "Richard Murray" 2:254/86.1@Fidonet or if you are on the Internet, "rmurray@arcticbb.demon.co.uk".

Also accepted: submissions, letters, stories etc etc...