

Feature List

FreeWorld BBS Feature & Idea List

Core System:

- written in C with portability in mind (Intel, Alpha, Sparc)
- runs as daemon/server process
- parse config files
- handles logging
- master/slave support for multiple BBS servers across a network to function as a single BBS
- ability to simultaneously handle 1000's of users across multiple servers
- dynamic shared object (DSO) support for modules

1. main/parent process initialized as root by init
2. main config file parsed / environment initialized
3. start <n> child servers as bbs.bbs
4. child servers wait for connect from login modules
5. login module sends user info to parent server
6. first available child attaches to calling terminal/port
7. client type determined (ansilconsolehttpdlwinl???)
8. appropriate client DSO module loaded into child
9. top menu file parsed
10. other dso modules can be loaded when needed based on menu tree
11. when user logs off, child process re-enters pool or dies

Login Modules:

- handles logins from various sources
- module for console/telnet/dial-in logins
- httpd module for web based logins
- win/X client daemon (standalone or inetd) for client logins
- each option configurable by the sysop

1. user logs onto the system as bbs or starts client software
2. determine login type (console|dial-in|telnet|http|client)
3. determine client type (console|ansil|ascii|html|vt100|client)
4. if user not new, call server and pass config/env data to it
5. if user is new, call server as new and pass config/env data to it
6. exit

BBS Getty:

- optional for those that do not wish to use other getty (mgetty, agetty, etc)
- capable of handling inbound/outbound fidonet calls
- fully configurable
- ability to answer bbs calls and pass fidonet calls to ifcico
- ability to be called from mgetty for fidonet calls

- can determine if call is standard dialup or client dialup

I currently have HylaFAX running on my system and have faxgetty attached to ttyS0 and ttyS1. If the call is a fax, it handles it. Otherwise, it is passed to mgetty as a direct connection. Mgetty then determines if the call is PPP, UUCP, FIDO, or regular dial-up login. I wish to keep this configuration and allow dial-in users access to the BBS by having them login as bbs then as their actual login. However, if I add more lines I do not need this setup and could attach bbsgetty to the additional serial lines to handle the calls directly.

HighLevel UI API:

- unified development environment independent of client type
- menus, windows, popups, scrolling, pulldowns, key control, dialog, buttons, etc
- client UI modules will implement specifics for the client type
- should be written in C++ with portability in mind
- provides high level display interface functions / widgets
- used for internal user interfaces, reader/editor, addons, etc
- progs using this library should be able to work on all future display modules with only the need to update external display file of it exists

This needs to be developed in parallel with the core system so that its functions are available for use within the BBS immediately. Since its the actual client modules that interperet the high-level UI code into the appropriate display, the ANSI/VT-100/Text clients can/should be worked on along with the UI. Other clients like httpd/X/Win95/etc can be done later after the UI has evolved a bit more.

Client UI Modules:

- unified menu files
- implements specifics for high-level UI functions for each client
- sets defaults based on term & client type client
- module specific configs
 - ◆ *ANSI/DialUp Module:*
 - ◇ general plain text and ansi interface
 - ◇ optionally display .asc, .ans, and/or .avt files
 - ◇ provides generic display when no .(asclanslvt) files exist
 - ◇ provides ansi/text specifics to high-level UI
 - ◆ *VT100/telnet Module:*
 - ◇ similar to ANSI/DialUp Module, but handles primarily telnet connections
 - ◆ *HTTP Module:*
 - ◇ interface into web server (cgi/ssi/apache module)
 - ◇ provides html specifics to high-level UI
 - ◇ uses .html files in the same manner as .ans and .asc
 - ◆ *Windows Client Modules:*
 - ◇ unified interface regardless of window system (MS, X, OS/2, ???)
 - ◇ dial-up and tcp/ip capabilities
 - ◇ .client files can be used like .ans and .asc files to control UI

Once the high-level UI is functional, any number of additional client display modules can be developed.

Security:

- bbs progs run as bbs.bbs
- admin progs run as bbs.bbsadmin
- users need bbs group privs
- admins need bbsadmin group privs
- unsigned short user and admin security levels (0-65535)
- unsigned user and admin security flags (32 bits each)
- uses system /etc/passwd & /etc/shadow files
- PAM support

User has standard /etc/passwd and/or /etc/shadow entry. The actual login name can be used as an alias in the BBS while the full name is also contained in the entry. All BBS users initially login as bbs which has the bbs login module as its shell. The login module prompts for the users actual login. Within the BBS environment, additional security settings can be configured for each user like in the old DOS BBS's. These settings can then be used to determine what the user has access to within the BBS. If the user also has access to a shell account, then they can either access the shell through the BBS interface, or login with their actual username instead of bbs initially. If the user has no shell account, then shell in /etc/passwd can be set to /dev/null.

Reader/Editor:

- uses high-level UI
- displays appropriate header fields when called for nntp/smtp/echo/net
- plain text editor with wordwrap, search/replace, etc
- pgp interface
- ispell interface
- handles taglines and/or signatures

From the users perspective, when reading or posting messages using the reader/editor e-mail, newsgroups, echomail, netmail, local messages all appear the same. If posting, then header fields are given based upon the type of message the user is posting.

NNTP Module:

- uses local news server or remote nntp server
- read/post privs configurable on per newsgroup basis (wildcard, first match)
- read/post privs based on security level and security flags
- uses internal BBS reader/editor
- optional news headers can be configured by sysop

This module provides an interface to a news server either on the same host as the BBS or a remote host. It is used to determine what security level is required for reading/posting to various newsgroups, and to what newsgroups are available to the BBS.

Here's a sample config file:

```
#read  post  flags  newsgroups
10     100    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX111  comp.bbs.unix.name
100    100    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX    comp.*
65534  65534  00000000000000000000000000000000  alt.porn.*
```

SMTP Module:

- uses local smtp server or remote relay
- read/write privs based on security level and flags
- uses internal BBS reader/editor
- optional headers configurable by sysop

This is similar to the NNTP module.

FIDO Module:

- should be sysop configurable for internal fidonet or via nntp
- internal message base support like most DOS BBS's
- use JAM/squish message base (Hudson has builtin size limitations)
- read/write privs based on security levels/flags
- multiple zone capabilities
- uses internal reader/editor
- can use bbsgetty for mailer
- optionally can be configured to get in/out mail from ifcico spool
- filebone handling (tic)

Chat Module:

- text/ansi/vt100/etc interface similar of DOS chat doors cosmetically
- multi user conferencing (perhaps an interface to local IRC server)
- user to user (free, busy, dnd, invisible on per user basis)
- user to all (user can disable 'all' messages except from sysop or system)
- user to sysop (privs configurable like most DOS bbs's)
- sysop to user (can't be overridden)
- sysop to all (can't be overridden)
- can play sounds on remote and sysop console (.ans, .mid, .wav, etc)
- java interface into cgi module ???
- can be used to interface with IRC

Whois Online Module:

- uses high-level UI
- interface to cfingerd or other finger server
- configurable so system users are invisible to bbs-users
- .plan, .pgp, .bbs_profile
- user configures what info others see
- sysop sets globals (user can't override)
- configurable security

Door Module:

- uses door.sys, doorinfo, etc
- interface to DOS doors via dosemu & high-level UI
- lan client/server interface to DOS box ??? (Caldera Open DOS???)
- provides stdio interface to unix programs via high-level UI
- security on a per door basis
- provide shell interface to BBS users with seperate shell account
- handles keyboard translations

This is what will give the BBS user the ability to either haccess his shell account from within the BBS or run Unix/DOS programs from within the BBS. For DOS doors, dosemu can be used or perhaps a LAN connected DOS box itself. All possible drop files should be supported for compatiliby. This same module can provide a stdin/stdout/stderr interface to unix applications.

File Module:

- uses high-level UI
- generic default area configuration for entire trees of dirs and files
- per area / per file access/read/write/upload/download configuration
- autoextract file_id.diz type files
- read files.bbs files
- virius scan on uploads (I think McHaffe <SP> has a virius scanner for unix)
- uploads without description require uploader to write one
- uses reader/editor module to read/write text files & write file descriptions
- name mangling support (displays full file name but will d/l in 8.3 form)
- archive conversion
- dive into archive files
- ability to semi-intelligently make a file description based on file contents automagically for files without descriptions. (sed/awk/regex)
- uses (xlyls)modem or any other external protocol for dial-up/telnet
- uses sendfile or ftp or http for for telnet/httpd/client

Here's another area to go to the extreme. I have several GB of files with some subdirs going 12+ levels deep. Some of the files are from my old DOS BBS file area with file_id.diz files in the zip. Others are my sunsite archive with .lsm files. Others have files.bbs files and others have nothing. If I were to have to attempt to manually configure all my file areas into the BBS, it will take me probably 10 years to complete. Therefore, what I'd like to see here is the ability to give the root directory for my files, and have the module build the index. If there are files I want to restrict access to, they can go in a restricted subdir and internal BBS security features can be used. Areas can be created based on subdir names. An interface fo find can be implemented to do RegEx search on file names/dirs and possible an interface to isearch (a text searching package). Also, a magic file can be used for determining file types.

Admin Interface:

- uses high-level UI
- all config files are plain text editable
- config files also be in database format
- curses/slang interface and X interface
- realtime monitoring windows (either use x-term or attached to tty)
- sysop take over/snoop mode (lets not turn this idea into a flame war)
- remote admin capabilities
- multi server systems can use same or seperate configs (globals are same)
- formats log files for simple reading

Misc System Features:

- perl and/or TCL as internal scripting language
- pre configured generic Q&A module
- access to most system, environment, user variables
- most anything that perl can do
- most system, environment, user variables available to all UI modules for inclusion in display files and menu options
- menu structures can be built like DOS systems or complex menu structures can be built using perl
- QWK/BlueWave (netmail, echomail, email, and usenet) support
- SQL (postgreSQL/MySQL) interface

Statics Module:

- top callers
- downloads/uploads
- fidonet/newsgroups stats
- all those other things provided by DOS bbs's and addon doors
- can be created using perl scripts
- uses UI library to generate displays

New User Module:

- called by login module for unknown users
- obtains user info automagically from client software
- gets initial user info and creates account
- switches user offer to new account and calls appropriate UI module

User Info Module:

- sysop sets global options based on security level
- some global options can be overridden by user based on security
- sets display options
- set up pgp key
- configure areas to show new files
- anything else that should go here
- configure newsgroups/echomail areas to scan for new messages

Display Files:

- filename = <name>.(asclanslavtrip|bhtml|html|client)
- if no display file exists when called, either an error will occur or the default menu display functions will be used
- .asc = plain text
- .ans = ansi display
- .avt = avatar display
- .bhtml = html files with embedded BBS display variables
- .html = standard html file
- .client = client display config (similar to fvwm config using UI API)
- if non client user has FancyDisplay=1 then .client file is translated by dial-up/telnet/console UI client.

Some sources to look at for ideas:

- apache httpd server - for server functions
- mc (Midnight Commander) - for file database module
- fidogate & ifmail - for fidonet code

Let's build a World Class BBS that will get World Class attention!