Lifespan Machine Software 2.0 : Installation Guide

Nicholas Stroustrup. August 3rd, 2017

nicholas_stroustrup@crg.eu
Centre for Genomic Regulation
Barcelona, Spain

This guide provides an overview of the Lifespan machine software and the computing environment on which it runs.   A step-by-step guide is provided for the installation, configuration, and installation of this software on a standard PC.   In steps involving multiple computers, the specific computer and operation system involved is specified italicized parentheses.

Table of Contents

Overview of Software Components

The Lifespan Machine provides an automated means for performing lifespan experiments on *Caenorhabditis* nematodes. The lifespan machine operates continuously over multiple weeks, acquiring and interpreting images of nematode populations. Routine execution of such a task in a research setting requires a persistent imaging platform robust enough to withstand variety problems including network outages and scanner errors, and smart enough to handle such errors in a way that does not compromise data quality. The lifespan machine accomplishes this through the combined efforts of four software components.

The first component is an ***image acquisition server*** (often shortened to ***"acquisition server"***) runs under Linux on a dedicated PC. This PC is directly connected to scanners via USB cables. A single computer running an image acquisition server can manage up to 20 scanners. Multiple computers each running an acquisition server can be operated simultaneously, allowing very large numbers of scanners to be operated simultaneously.

The second component is an ***image processing server*** (also called the ***image analysis server***) which is designed to run on many types of computers— a personal laptop, a lab desktop workstation, or a high performance computing cluster. Precompiled binaries are available for windows and the software can be compiled under Linux. Multiple image processing servers can be run in parallel on the same data set, allowing rapid image processing. The image analysis software is designed to run persistently in the background, allowing images to be automatically analyzed as they are acquired.

The third component is a GUI client program called *the **worm browser***. Precompiled binaries are available for windows, and the software can be compiled under Linux. This program is used directly by the experimenter, for example on their personal laptop, allowing them to schedule experiments, generate data files, and validate data collected by the lifespan machine.

The fourth component is the ***lifespan machine web interface*** which is a set of PHP scripts running on any server, usually the machine hosting the image acquisition server. The web interface provides access to experiment information and allows image processing jobs to be scheduled. Users access the web interface scripts through a web browser. The web interface is served by a machine on the lab's local network, usually the same hardware that runs the image acquisition server.

All three software components communicate through a single, **central MySQL database**, usually hosted on one of the image acquisition servers or on an institutional server. This database contains metadata describing collected image data, and also coordinates the automated analysis of images. All three components require access to a **central file server,** where image data is written and retrieved. This is usually a departmental or university file server, but lab-specific Network Accessible Storage (NAS) can be set up in cases where pre-existing IT infrastructure is lacking.

The image acquisition server must be run on either Scientific Linux 7.3 (recommended) or Scientific Linux 6.8.   The image analysis server can run either on Scientific Linux 7.3, Scientific Linux 6.8, or Windows 7/8/10.   Scientific Linux is a free Redhat variant maintained by Fermilab and CERN. Precompiled binaries are available for Windows. The image acquisition server needs to be compiled *in situ* on its Linux server. All three software components of the lifespan machine are written in C++.
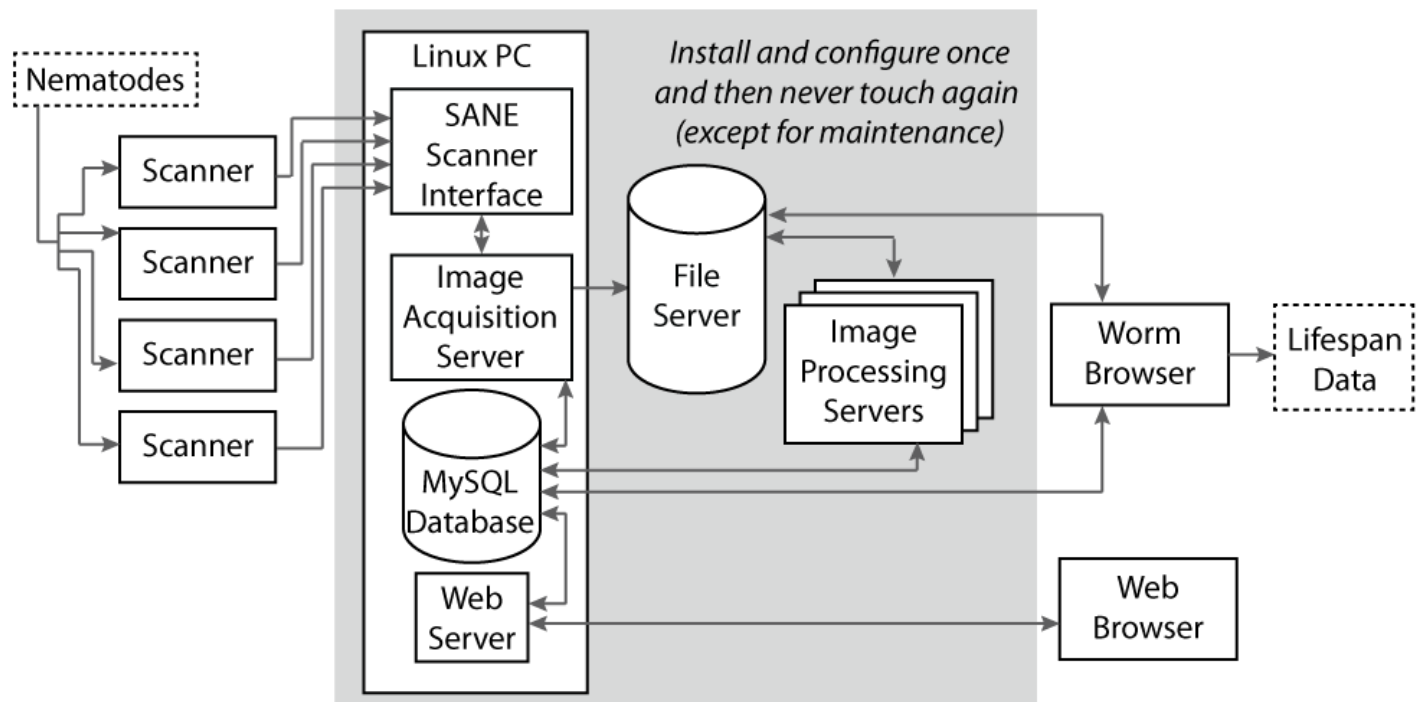
Figure 1—Lifespan Machine Schematic. The lifespan machine collects images of nematodes and interprets them to estimate the lifespan of each individual. This is accomplished through interaction of several hardware and software components, as shown. It is important to note that routine use requires an experimenter to interact only with scanners (to load animals), the worm browser (to schedule experiments and validate automated results), and a web browser (to supervise the process and schedule image processing tasks). Most users need not understand all the details of these components, but someone needs to be available to solve problems when they arise. Problems are most frequently encountered at the very beginning of an experiment, when animals are loaded into scanners and the experiments are set up. This document details the installation and configuration of all components shown above, excepting scanners whose configuration is detailed elsewhere.

Required Computing Infrastructure
- A PC running the Image Acquisition Server
  - At least 2 Gb of RAM
  - Any x86 processor made after 2010
  - A USB 2.0 port for direct connections to scanners

- o Scientific Linux. The 64bit version of Release 7 is highly recommended; 6.8 will probably work with minor alterations to the installation procedure.
  - o A hard drive large enough to hold buffered images (e.g. >500Gb)
  - o The server will host a network-accessible MySQL database and apache web server.
- A PC or high performance computing cluster node on which to run the image processing server
  - o Linux 7 (recommended), 64bit Windows 10 (recommended), or Linux 6.8, or 64Bit versions of Windows 7 or Windows 8
  - o Any x86 processors made after 2010. Four core Intel i7 processors can run eight image analysis server nodes simultaneously, speeding up analysis.
  - o At least four Gb of RAM. (16 or more GB recommended)
  - o A fast hard drive (e.g a SSD) speeds up analysis

- A long-term image storage location accessible over the network, for example an institutional fileserver, a stand-alone NAS attached to the local ethernet, or a directory on the machine hosting the image acquisition server shared via Samba or CIFS. All captured images will be stored here, so you'll want at least two hundred and fifty GB per four week lifespan performed across 10 scanners. Note that at 2013 prices, one terabyte of storage costs less than the plates required to run the experiment itself. The storage location must be accessible by the image acquisition server, the worm browser, and the image processing server.

An important Note about USB Cables and Hubs

Many vendors will sell long (10+ feet / 3+ meters) USB cables. These generally provide very poor performance, working fine immediately after installation but failing a few days or weeks later. Failure of a cable will not produce an outright disconnection of scanners, but instead will produce a slow degradation in the reliability of the system, with scanners mysteriously failing in the middle of experiments. Use short cables.

USB hubs themselves age and die—we have found our hubs become unreliable after a few years, with USB problems arising on previously stable installations. Cheaper hubs will on average fail sooner or not work at all. We are currently looking into industrial-class USB hubs to evaluate whether they are worth the additional cost; contact Nicholas Stroustrup to get the latest recommended version.

Installing Scientific Linux on the Image Acquisition PC *(Acquisition Server; Linux)*

The user must make informed decisions on the details of their Linux server based on the specifics of their local environment. A linux system is a complex machine and there are many different ways to configure each aspect. depending on personal preference and institutional policy. This installation guidelines provides a step-by-step guide that, if followed carefully, will work as described in the methods paper TBD, based in part on the 2013 Nature Methods paper "The C. elegans Lifespan Machine".

1. Download the latest version of Scientific Linux 7 and burn it onto a DVD. Boot the computer on which you're installing linux using the DVD. The installation procedure should run automatically. Linux 7 offers several pre-configured "base environments"; this can be altered later on, but for now choose "Development and Creative Workstation". To speed up the installation process, you can select "add-ons" for the environment. Add
   - o Additional Development
   - o Development Tools
   - o E-mail Server
   - o Emacs
   - o File and storage server

- o   Hardware monitoring Utilities
- o   MariaDB database server
- o   Network File system client
- o   PHP support

2. When asked, create a user account with the name "image_server" (do not include the quotation marks). We will use this account to store all the lifespan machine code. You can use any name you prefer, but we will refer to it as "image_server" in the remainder of the document.

3. Choose all other installation options as per your personal preference—they should not alter the lifespan machine's functionality.

4. Make sure that the network connection is enabled (this is usually done via an icon on the Linux desktop in the top right menu bar)

5. Allow the ports through the machine's firewall using scientific linux graphical firewall configuration tool, located in the main menu option Applications/Sundry/Firewall
   - o   Allow the following ports (sometimes called "zones / trusted services")
     - ▪   ssh
     - ▪   samba
     - ▪   www
     - ▪   https
   - o   Allow the following additional ports under "Ports": mysql, port 3306 (both tcp and udp), and the VNC ports 5900 and 5901.

Configuring everything to run the image server *(Acquisition Server; Linux)*

Configuring the Mysql Database

Set the database to always run by typing *systemctl enable mariadb*

Run the database by typing *systemctl start mariadb*

1. Set the mysql root password by typing *mysqladmin -u root password 'your_root_password'* replacing your_root_password with the password you want, keeping the quotation marks.

2. Log into the MySQL database by typing *mysql –u root –p*

3. Create a user for the image server by typing *CREATE USER 'image_server'@'%' identified by 'yourpassword';* Again, replace yourpassword with another password, keeping the quotation marks.

4. Create the same user account to be accessed on the local machine by typing *CREATE USER 'image_server'@'localhost' identified by 'yourpassword';* Use the same password as the previous step, keeping the quotation marks.

5. Create the central database for the image server by typing *CREATE DATABASE image_server;*

6. Create the local buffer database by typing *CREATE DATABASE image_server_buffer;*

7. Grant access permissions by typing *GRANT ALL on *.* TO 'image_server'@'localhost';*

8. Grant additional permissions by typing *GRANT ALL on *.* TO 'image_server'@'%';*

9. Exit the mysql client by typing *exit;*

10. Configure the my.cnf file to have more sensible memory use parameters. This file is located either at /etc/mysql/my.cnf or /etc/my.cnf. Open it for editing by typing the command
    emacs /etc/my.cnf
    The defaults are much too conservative in memory use and caching, leading to terrible performance. In the emacs window, scroll down to the bottom of the file and add the following lines
    innodb_buffer_pool_size = 512M
                 key_buffer = 1024M

query_cache_size = 8M

query_cache_limit = 256M

thread_cache_size = 8

innodb_buffer_pool_size = 2048M

innodb_flush_log_at_trx_commit=2

table_cache=1024

thread_cache=16

query_cache_size=128M

These values work well on a machine with 4GB of RAM. Machines with more should use large values—more information about tuning mysql databases is available from many online sources.

11. Install the mysql database schema by typing the command

    mysql -u root -p image_server < /home/image_server/lifespan/files/image_server_db_schema.sql

12. Restart the database by typing *systemctl restart mariadb*

Configuring the httpd web server

1. Set the web server to always run by typing *systemctl enable httpd*
2. Run the web server by typing *systemctl start httpd*
3. If httpd doesn't appear to work correctly, one common problem is that the server name is set incorrectly. There is a line in the http.conf file (usually located in the directory /etc/httpd/conf/httpd.conf) that specifies the ServerName variable. It should be changed to ServerName localhost
4. Allow PHP to access the mysql server. Open a terminal window, and type the command

   setsebool –P httpd_can_network_connect=1
5. Allow PHP to access a cifs drive by typing the command

   setsebool -P httpd_use_cifs=1

Download and install the Worm Browser and Image Analysis Server *(Analysis Server; Windows)*

The image analysis server can run on either Linux or Windows. On Windows, Worm Browser and Image Analysis Servers are distributed as executable binaries, available from the lifespan machine software repository hosted on github.com.

There are two methods to obtain these executables. The preferred method is to download the windows github client program from https://desktop.github.com and use it to download the lifespan machine git repository (following the instructions specified in section " Linux and Windows: Downloading the Lifespan Machine Software Repository" of this document). This method requires a more work initially, but is better in the long term as it makes it very easy to install updates and bugfixes.

The second method for obtaining the lifespan machine executables is to download them directly from the github.com website.

The version 1.x executables are available at https://github.com/nstroustrup/lifespan/tree/master/binaries/windows

The version 2.x executables are available from https://github.com/nstroustrup/lifespan/tree/flow/binaries/windows

Also included in the binaries/windows directory a set of DLL files that must be downloaded be placed in the same directory as the executable files: libmysql.dll, openjp2l.dll, and xvidcore.dll.   On systems where the Intel Performance

Primitives do not function out of the box, (See the section "Installing the Intel Performance Primitives" later in this document), a special binary and additional 29 dll files must be downloaded and placed in the same directory as the executables, located in the directory binaries/windows/ipp_dlls .

On windows, software installation is easier as nothing needs to be compiled. After downloading the executables, skip to ns_image_server.ini and ns_worm_browser.ini. When each program is downloaded and run for the first time, the two configuration files are created automatically, each including a description of the configuration parameters.

This dictates that the acquisition server be installed first. Note: the image analysis server is designed to run many instances in parallel, to allow fast processing of images. This is set in the ns_image_server.ini option nodes_per_machine. On Intel Core Duo, i3, i5, and i7 processors, the best results are obtained when nodes_per_machine is set either to twice the number of processor cores on the machine, or twice the number of GB of memory, whichever is smaller.

The Worm Browser and Image Analysis Server can also be compiled from source using the provided Microsoft Visual Studio 2015 solution and project files.

Installing additional software packages *(Acquisition Server Linux)*

1. If you are new to linux, check out the appendex Appendix 1: Useful Linux Commands" and read up a bit on how the command line works.
2. Many software packages must be installed as a prerequisite for compiling and installing image acquisition server. The easiest way to do this is using the command line tool "yum". To do this, you should open a terminal window (it is the first item in the main menu "Applications"). Installation requires administrator access, so you need to log into the root account, whose password you set during the scientific linux installation. To log into the root account, in the terminal window type *su root* and press enter. Type in the root password (the characters you type will not be shown on the screen to keep them secret) and hit enter. If you do not receive an error message, you are now logged into the root account.

   To install each package, you must type the command *yum install [package-name]* replacing [package-name] with the name of each package. Do this for each package below. For example, to install gcc, type *yum install gcc* and then hit enter.   You may have to type *y* to confirm the installation. Many of the packages may already be installed, which is good.

- gcc
- httpd (the apache web server)
- libtiff-devel
- libjpeg-devel
- freetype-devel
- mysql-devel
- mariadb-devel
- php
- php-mysql
- libusb-devel
- libusb-static
- cmake
- libpng
- zlib
- git
- openjpeg-devel
- libtool
- autoconf
- automake
- libusb1-devel (NOTE: different than libusb-devel)
- xorg-x11-server-devel (for linux worm browser)
- mesa-libGL (for linux worm browser)
- freeglut-devel (for linux worm browser

3. The current SANE version, 1.0.25, bundled with Linux 7 does not fully support Epson v800 scanners. If you are using the v800 model scanner, make sure the default SANE scanner package (sane-backends and sane-backends-devel) are uninstalled. You can do this by typing the two commands *yum remove sane-backends* and *yum remove sane-backends-devel*

Updated drivers are included in the lifespan machine git repository, and will be installed in the next section.

Downloading the Lifespan Machine Software Repository *(Acquisition and Analysis Servers; Linux and Windows)*

The source code for the lifespan machine code is stored in a git repository on github.com . The most convenient means for accessing the code is using the git commandline client. Using this client will allow you to quickly download and install future software updates and bug fixes. If for some reason you cannot use the git client, it is also possible to manually download the source code directly from github.com, but this is no longer recommended as it makes subsequent updates to the code more difficult to obtain and install.

1. Open a terminal window, and navigate to the home directory of your image_server user account. The command to do this is *cd /home/image_server*
2. Download the lifespan machine git repository from github.com by typing the command
    git clone https://github.com/nstroustrup/lifespan.git
3. Switch to the latest version of the software (called "flow") by typing the command
    git checkout flow

Installing extra libraries included in the lifespan machine source code repository *(Acquisition and Analysis Servers; Linux)*

Enter the external compile libraries directory by typing
cd /home/ image_server/lifespan/external_compile_libraries

1. Type the command *ls* . This will list a set of libraries, each in their own subfolder. You need to compile and install each library, one at a time.
    o **xvid :** In order to compile videos of captured image data, you need to install the xvid video encoder.
        1. Enter the xvidcore directory by typing *cd xvidcore-1.3.4*
        2. Enter the linux source code directory by typing *cd build/generic*
        3. Generate the configure scripts by typing *sh ./bootstrap.sh*
        4. Configure the compilation by typing *./configure*
        5. Compile the source code by typing *make*
        6. Install the source code by typing *make install*
        7. In some cases, the xvidcore installation scrips do not properly register all of its shared libraries. If in later steps you encounter problems while compiling the lifespan machine source code, run the following commands
            • *cd /usr/local/lib* (or wherever you set xvidcore to install)
            • *ln –s libxvidcore.so.4.3 libxvidcore.so*
            • *ln –s libxvidcore.so.4.3 libxvidcore.so.4*
    o **Fltk:** if you want to build the worm browser under linux, you need to install this GUI library. Most users run worm browser on windows, in which case this not necessary.
        1. Return to the external_compile_libraries directory
            Enter the fltk directory by typing *cd fltk*
        2. Configure the compilation by typing *./configure --enable-shared*

3. Compile the source code by typing *make*
4. Install the source code by typing *make install*

o **dmtx:** This is a barcode reading library used to identify scanners.
1. Return to the external_compile_libraries directory
2. Enter the dmtx directory by typing *cd libdmtx-0.7.4*
3. Configure the compilation by typing *./configure*
4. Compile the source code by typing *make*
5. Install the source code by typing *make install*

o **openjpeg2000:** The lifespan machine uses jpeg2000 compression. The latest version of the openjpeg library, version 2.1.2, is not included in linux 7, so we install the latest code here.
1. Return to the external_compile_libraries directory
2. Enter the openjpeg directory by typing *cd openjpeg-2.1.2*
3. Configure the compilation by typing *cmake .* (note the period; it is necessary)
4. Compile the source code by typing *make*
5. Install the source code by typing *make install*

o **sane-backends:** The lifespan machine uses SANE scanner drivers.
1. Return to the external_compile_libraries directory
2. Enter the sane directory by typing *cd sane-backends-1.0.27*
3. Configure the compilation by typing *./configure*
   or, If you only plan to use Epson scanners, you can speed up installation by typing
   ./configure BACKENDS="epson2" RELOADABLE_BACKENDS="epson2"
4. Compile the source code by typing *make*
5. Install the source code by typing *make install*
6. In many environments, you will need to explicitly enable SANE support for the v800 scanner. To do this, you must copy two files to two SANE configuration directories. The following commands assume a default location for these SANE directories; they may vary on your system and you should change the following commands accordingly.
   Type the command
   cp ../../files/dll.conf /usr/local/etc/sane.d/dll.d/
   and then type the command
   cp ../../files/epson2.conf /usr/local/etc/sane.d/

Installing the Intel Performance Primitives *(Acquisition and Analysis Servers; Linux; Windows)*

To run faster on various hardware architectures, v2.0 of the lifespan machine image analysis software uses Intel's Integrated Performance Primitives. On Linux, the library binaries must be downloaded from the intel website. Version 2017 can be downloaded at no cost from https://software.intel.com/en-us/intel-ipp . A copy may also be available from Nicholas Stroustrup upon request. On Windows systems, in most cases IPP does not need to be installed. When IPP does not function correctly, a special executable and a set of IPP dlls must be used, included in the github directory binaries\windows\ipp_dlls.

Installing the image server software *(Acquisition and Analysis Servers; Linux)*

1. The image acquisition server must be compiled and installed.
   - Type *cd /home/image_server/lifespan/build*
   - Configure the compilation by typing *cmake .* (note the period; it is necessary)

- To install the worm browser in linux, type *cmake -DBUILD_WORM_BROWSER .* (note the period; it is necessary)
- Compile the source code by typing *make*
- Install the source code by typing *make install*

Mount a Network Accessible Storage directory for long-term storage of images *(Acquisition and Analysis Servers; Linux)*
NAS directories can be mounted by adding lines to the /etc/ fstab file. On some systems, an alternative is to use the autofs package, but the author does not have direct experience with this. To use the fstab file, first, create the desired mount point in the /mnt directory. For example /mnt/fontanalab

1. The following line is an example that instructs the machine to automatically mount the CRG fileserver

//isis2/users/xxx/microscopy   /mnt/isis2   cifs credentials=/root/.cifspass   0 0

2. The .cifspass file will be used to automatically provide credentials to the file server. The .cifspass is a simple text file with two lines:
   username=myusername
   password=mypassword
3. Mount the network folders via the commands "mount /mnt/myfileserver"

Install and Configure the image server web interface *(Acquisition Server; Linux)*
1. The web interface consists of a set of PHP scripts. These are located in the website subdirectory of the git repository. They need to be copied to the linux directory used by the web browser. This is located at /var/www/html . This can be done automatically by running the script located in the source code directory (the directory created when you downloaded the code from github)
   /home/image_server/lifespan/ns_install_website.sh
   Or it can be run manually by typing in the command
   cp –r /home/image_server/lifespan/website/ /var/www/html*
2. This web interface has a configuration file called ns_image_server_website.ini . An example configuration file is included in the git repository. In the previous step, you copied it to the path
   /var/www/html/image_server_web/ns_image_server_website_template.ini
   You can use this template to create a configuration file specific to system. Typing the command
   cp /var/www/html/image_server_web/ns_image_server_website_template.ini /var/www/html/ image_server_web/ns_image_server_website.ini
3. Edit the new configuration file that you created by typing the command
   emacs /var/www/html/ image_server_web/ns_image_server_website.ini
   Specify values for each configuration option, in particular the mysql database, user, and password fields. Some values are explained in more detail in the next section "Linux: Setting up the web interface to show captured image data".
4. PPH by default hides error messages, instead displaying a blank page. To have the server output errors to the final user (allowing problems to be more easily debugged), change the following options in the php.ini configuration file, usually located in /etc/php.ini
   Edit the php.ini file by opening a terminal window and typing *emacs /etc/php.ini*
   In the emacs window, find the line of the file that specifies each field (display_errors, error_reporting, and date.timezone) and modify each to show the following value:
   display_errors = On

error_reporting = E_ALL & ~E_Notice

date.timezone = Europe/London

These lines may be commended out by a leading # . If so, you need to remove this character. If you are not in the London time zone, set the timezone to the appropriate one listed here

http://php.net/manual/en/timezones.php

5. Apply these changes by restarting the web server. Open a terminal window and type *systemctl restart httpd*

Setting up the web interface to show captured image data *(Acquisition Server; Linux)*

The image server web interface can be set up to display images captured by scanners, making it easy to browse through collected imagery. The challenge here is that image data is often stored on a networked drive, mounted outside the standard directory for web data: /var/www/html . The web server needs to be configured to access the images at their location.

1) First, you must create a symbolic link in the website directory tree that will allow the web server to access data in the long term storage directory. This will most likely be the folder you mounted in the earlier section of this document "Linux: Mount a Network Accessible Storage directory for long-term storage of images. To create a symbolic link, open a terminal window and type.

   ln –s /mnt/myfileserver /var/www/html/long_term_storage

   Replace "myfileserver" with the actual name you used for this directory.

2) By default, the apache web server follows symbolic links. However, on some installations this may need to be set explicitly, by editing the web server configuration file, httpd.conf , adding an option FollowSymLinks . A google search can provide more detail. This is often located in the directory /etc/httpd/conf

3) Edit the ns_image_server_website.ini file located in /var/www/html/image_server_web by opening a terminal window and typing emacs *</var/www/html/image_server_web/ns_image_server_website.ini*

4) The value of $ns_image_server_storage_directory should be set to the path you specified in step one of this section (e.g the subdirectory of /var/www/html where you placed the symbolic link). That means, if you created the symbolic link as /var/www/html/long_term_storage , $ns_image_server_storage_directory should be set to "/long_term_storage".

5) It is possible that selinux might disable access by the web server to the long term storage directory. One possible solution would be to enter the command

   setsebool -P httpd_use_cifs on

Set the image server to run at startup (optional) *(Acquisition Server; Linux)*

The image server can be configured to run automatically at startup. This is useful if, for example, there is a brief power outage in the middle of the night.

1. The rc script named /files/ns_image_server_rc_script needs to be transferred to the directory and renamed /etc/rc.d/init.d/ns_image_server . This can be done automatically by typing the command

   ./ns_install_startup_script.sh

2. Go to /System/Administration/ Services and set ns_image_server for runlevel 4 and 5.

Configure the image server software *(Acquisition and Analysis Servers; Linux and Windows)*

The lifespan machine must be configured to interact correctly with your SQL database and your file server. A variety of important configuration options are specified in a file located on each machine running image server software. This file

does not have to be written from scratch; when the image server or worm browser is launched for the first time, it will create a template ns_image_server configuration file.   On Linux systems, the default location for this will be /usr/local/etc /ns_image_server.ini , though on some systems the location may be different. On windows systems, the default location is the same directory as the program executable, but on some systems the software will create the directory c:\server and store the configuration files there. Image server software components, when launched for the first time, will output a message telling you where this file is located.

Once created, the configuration file can be modified with a text editor to specify a variety of important options. Users should read through the ns_image_server.ini file to understand the options that are documented there. A subset of important options are described below. ***Set each of these options before running the lifespan machine:***

host_name: Each instance of the image acquisition and image analysis servers needs to have a unique name to identify it. Thus, host_name should be set to a different value on every LINUX or Windows machine running the software. Use a name that you'll recognize, such as linux_server_on_my_desk, bob, or lab_desktop_1

long_term_storage_directory: All image server software must be able to access a central directory used to store images. This is often located on a NAS or institutional file server. This directory should be mounted as a path on the machine running the server. You likely set this up in the previous section of this document, "Linux: Mount a Network Accessible Storage directory for long-term storage of images".

results_storage_directory: All image server software must be able to access a central directory used to store processed statistical data, including survival curves, descriptions of worm movement, etc. This is often located on a NAS or institutional file server. This directory should be mounted as a path of the machine running the server. Set this parameter to the location of that directory.

volatile_storage_directory: The image acquisition server and image analysis servers need to store temporary files on the local machine. Set this parameter to the location of that directory; it can be anywhere you like. For image acquisition servers, this is the local buffer for captured images pending transfer to the long term storage directory, so you should locate the directory on a drive with a couple hundred 100 GB of free space.

Access to the central SQL database
These parameters need to be set to match the account set up on your central sql database, to allow the server to log in.

central_sql_hostname: The IP address or DNS name of the computer running the central SQL server. On the linux server, this will be localhost. On other machines, this should be the ip address of the linux server.

central_sql_username: The username with which the software should log into the central SQL server

central_sql_password: The password with which the software should log into the central SQL server

central_sql_databases: The name of the database set up on the SQL server for the image server. It's possible to specify multiple independent databases, each separated by a semi-colon, but this is not needed in simple installations.

Access to the local SQL database
Image acquisition servers use a local SQL database to store metadata pending its transfer to the central SQL database. This lets acquisition servers continue to operate correctly through network disruptions, sql database crashes, etc. These parameters need to be set to match the account set up on the machine's local sql database, to allow the server to log in

local_buffer_sql_hostname: The IP address or DNS name of the computer running the local SQL buffer. This is only needed for image capture servers, and in all but exceptional cases should be set to localhost

local_buffer_sql_username: The username with which the software should log into the local SQL buffer

local_buffer_sql_database: The name of the local SQL buffer database

local_buffer_sql_password: The password with which the software should log into the local SQL buffer

Image Acquisition Server Settings

These settings control the behavior of image acquisition servers

act_as_image_capture_server: Should the server try to control attached scanners? (yes / no). Set this to yes if you want to capture images on this server. Set it to no if you are only going to perform image analysis on this server.

Image Analysis Server Settings

These settings control the behavior of image processing servers

act_as_processing_node: Should the server run image processing jobs requested by the user via the website? (yes / no). Set this to yes if you want to perform image analysis on this server. Set it to no if you want to capture images—it is not advised to do both on the same machine.

nodes_per_machine: A single computer can run multiple copies of the image processing server simultaneously, which allows many jobs to be processed in parallel. Set this value to the number of parallel servers you want to run on this machine. This can usually be set to the number of physical cores on the machine's processor, or the number of GB of RAM on the machine; whichever is smaller.

Configuring the image server to generate videos *(Analysis Server; Windows)*
Running under windows, the image server can generate time-lapse videos of collected imagery. To do this, the image server software needs to access the encoding program x264, currently available at
http://download.videolan.org/pub/x264/binaries/

Most users will want to download the most recent release version of the 32 bit windows binaries, which is currently available at
http://download.videolan.org/pub/x264/binaries/win32/x264-r2431-ac76440.exe

Save this executable (.exe file) on the machine running the image server—it is easiest to put it in the same directory as the image server executable--and specify its filename in the ns_image_server.ini field "video_compiler_filename" . After a restart, the image server will then be able to process video creation jobs submitted using the web interface.

Configure the worm browser software *(Analysis Server; Linux and Windows)*
The image processing server is configured using the ns_image_server.ini that is created the first time you run the image server or worm browser. The default location for this file under windows is c:/server, or in the same folder as the executable. On linux, the file will by default be created in the directory /usr/local/etc/ns_image_server/ The worm browser has an additional configuration file, ns_worm_browser.ini, that will be created in the same folder, the first time the worm browser is run. Open this file in an file editor (notepad or emacs) to specify these values:

max_width: The maximum width of the worm browser window

max_height: The maximum height of the worm browser window

hand_annotation_resize_factor: How many times should images of worms be shrunk before being displayed when looking at storyboards? Larger values result in smaller worms during by hand annotation of worms

mask_upload_database: The SQL database in which image masks should be stored. This is usually set to the value specified for the central_sql_databases option in the ns_image_server.ini file

mask_upload_hostname: The host name (e.g bob or lab_desktop_1) of the server where sample region masks should be uploaded. This is usually set to the value of the host_name option set the acquisition server's ns_image_server.ini file

verbose_debug_output: If this option is set to true, the image server and worm browser will generate detailed debug information while running various steps of image acquisition and image processing. A file containing this output will be written to the volatile_storage directory.

Run the Image Acquisition Server *(Acquisition Server; Linux)*

1. Start the image server by opening a terminal window and typing *ns_image_server*. If the server detects a problem in its configuration, it will halt and display an error message. Correct the problem and try again, until the server starts correctly.
2. In some installations, the image server might report an error about not being able to access a log file. This might happen for two reasons:
    o The image server does not have access to the volatile storage directory set in ns_image_server.ini. This can be solved either by opening a terminal window using chmod to set the permissions for that directory correctly (google chmod for more info), or by running the image server under a root user account (type *su root* to switch to this account) We usually choose the latter option.
    o The image server is already running. There are a variety of ways to fix this, including killing the process or typing *ns_image_server stop*.
3. If the command ns_image_server is entered without any arguments, the image server will try to launch. However, a variety of options can be provided that will alter the image server's behavior. These options can be seen by opening a terminal window and typing the command *ns_image_server help*
4. If you want the image server to detect scanners and run experiments, make sure to edit the line in the ns_image_server.ini to specifiy act_as_image_capture_server = yes
5. This image server should never be stopped during image acquisition. To shut down or restart the server, open a separate terminal window and enter the command *ns_image_server stop* or *ns_image_server restart* . If no scans are running, you can also press CTRL-C to send the image server a SIGINT signal. The previous command line options are preferred to CTRL-C, as the latter may terminate ongoing scans, leaving scanners in an error state, requiring they be power-cycled.

Naming Scanners and Generating Barcodes

Each scanner needs to be assigned a unique name. These names allow the server software to distinguish between scanners, and also allows the user an easy means to keep track of the locations of each plate under observation. We tend to use short, memorable names, for example cori, axel, or john . Scanners names should be three or more characters long. We affix these barcode on the inside surface of the bottom scanner glass. An example is shown to the right.

sydney

These barcodes are standard dot matrix codes , and a tool has been made to easily generate them, ns_image_server_barcodes.exe. This windows commandline binary is available from the github repository, currently accessible at https://github.com/nstroustrup/lifespan/tree/master/binaries/windows

The command *ns_image_server_barcodes –c mybarcodes.tif robert linda gary*
will generate a single file, mybarcodes.tif, containing the three barcodes requested. These should be printed out, cut to size, and affixed to the bottom surface of scanner glass.
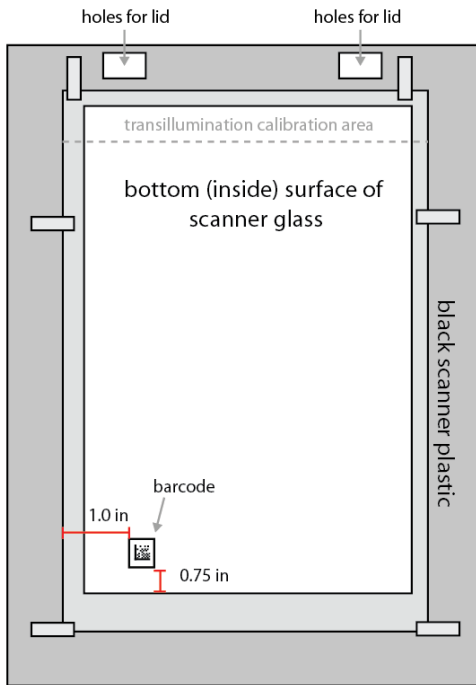
The barcodes can be affixed by printing them on standard paper and using rubber cement as glue. Other types of glue may work also, but rubber cement does not dissolve printer ink, and furthermore is very easy to remove from scanner glass if spilled or smudged. Alternately, barcodes can be printed onto thin adhesive CD label paper. It is important that the paper lie flat without bubbles to the bottom of the scanner, so that it does not get caught by the scanner bar as it moves past. Many types of adhesive labels are too thick to fit in the space available, which is why "light" CD labels tend to work best. In all cases, all four corners of the barcode are securely affixed, to prevent them from catching the scanner bar and destroying the label.

The barcode should be affixed such that it is scanned correctly within the region specified by the option device_barcode_coordinates specified in the ns_image_server.ini file. The default for this is "-l 0in -t 10.3in -x 8in -y 2in", meaning that the bar code should be within a two inch region 10.3 inches from the top of the reflective scanning region, and anywhere within 8 inches from the left side of the scanner surface. A diagram of this, taken from the perspective of inside the scanner, is shown to the left.

Getting Scanners Detected *(Acquisition Server; Linux)*
The image server interacts with scanners using the linux open source scanner driver framework SANE. SANE is extensively documented online, and handles all of the scanner-related hardware issues so that the lifespan machine doesn't have to. The basic protocol for getting the lifespan machine running is as follows

1. Plug a scanner into the computer via USB
2. Confirm that the SANE system has detected your scanner, by inspecting the results of the command sane-find-scanner. If your scanner is on this list, you're in good shape! This usually just works right out of the box, but if not, you'll need to trouble shoot. Is your scanner turned on? Plugged in? Have you set up SANE to use USB (this is the default behavior)? Are you using a supported scanner model? Check out http://www.sane-project.org. Confirm that your scanner hardware is being detected by the command sane-find-scanner . Note that in some places the v700 is referred to as gt-x900. The v800 is also in some places referred to as GT-X980
3. If your scanner is not detected, copy the epson2.conf and dll.conf files provided in the installation into their respective folders (See instructions in the "sane-backends" subsection of the "Installing extra libraries included in the lifespan machine source code repository" section of this document).
4. Confirm that you've correctly attached a barcode to the inside of the scanner surface, allowing the image server software to uniquely identify it. If you don't have a barcode yet, please refer to "Naming Scanners and Generating Barcodes".

5.   Restart the image server, or request a "hotplug" discovery of scanners via the web interface. The software should scan the barcode of each scanner and report back whether the scanner could be identified.

Security *(Acquisition and Analysis Servers; Linux)*

It is a poor idea to have your Linux server exposed to the public internet. Criminals looking to exploit systems have significantly more time and experience than the average researcher—the best way to keep your server secure is to keep it isolated within an institutional network, behind some type of hardware firewall. If this is not possible, a second option is to configure the machine to reject access from IP addresses outside the institutional network. There are many tutorials online describing how to do this, for example www.ghacks.net/2009/03/27/configure-a-linux-firewall-with-webmin/

It is possible to password protect the lifespan machine web interface. This can be done using the facilities provided by the apache web server. Specifically, password protection can be set up by placing an .htaccess file in the base directory of the lifespan machine web interface. There are many tutorials online on how to do this, for example http://www.linux.org/article/view/-htaccess-password-protection-securing-a-folder-in-a-website

A note about institutional IT departments

IT departments in research universities work hard to maintain complex services in a constantly evolving environment. This leads to a natural conservatism in access policies for network resources. The author of this document has had very good experiences working with IT departments to establish policies conducive to high-throughput automated microscopy. However, this experience may not be universally shared.

IT departments, for example, might be unhappy with having a web server running on their local network. Networked computers certainly should be forbidden from serving to the open internet, but skeptical IT staff should be encouraged to articulate the exact risks created by a properly firewalled local Linux-based web server.   Some IT departments might be hesitant to grant automated software access to their file servers. In one case, for example, an IT department was concerned enough about a lifespan machine installation to remotely terminate it, because it "opened too many files". However, the lifespan machine software is much better behaved and much more predictable than many organic operators.   IT departments generally become more understanding after the technology is explained to them.

A frequent suggestion raised by IT departments is that labs should set up their own network or network file server, disjoint from the campus network resources. For the IT staff, this is certainly the simplest solution, as it requires no reconfiguration of institutional resources. For lab members this also might sound like the simplest solution, as the task of building a new network might be faster and more enjoyable than advocating for change in an institutional IT policy. In practice, however, the Do It Yourself approach to network infrastructure is rarely the best solution for a research lab. Many unforeseen complications will emerge—for example, desktop computers will have to be connected to both networks. The lifespan machine will always be hard to access, isolated in its own network. Per byte, custom-built networks will be significantly more costly than large institutional network. Furthermore, the creation of a private network will force researches (usually Ph.D students or post docs in the life sciences) to shoulder the exact network administrative duties that IT departments exist to provide.

Users are encouraged to engage in a productive dialog with their institution's IT department and administration, to develop a network environment that is both safe and useful.

Upgrading the image server *(Acquisition and Analysis Servers; Linux and Windows)*

The linux image capture server and the windows data analysis binaries (image analysis server and worm browser) can be

updated independently. The image capture server running version 1.x (e.g. 1.7, 1.8 and 1.9) is compatible with image analysis severs and worm browsers running v2.0. This is useful, as it is often relatively easy for labs to update their windows image analysis software but difficult to update their linux capture server. Only the former needs to be done to benefit from v2.0 features.

Upgrading from 1.x to 2.0

The lifespan machine version 2.0 adds several features, including improved death time detection accuracy, faster image analysis, 5x smaller image file sizes using jpg2000 compression, and high performance computing cluster support. Version 2.0 uses a backwards compatible database format, meaning that both 1.x and 2.x versions of the software can run using the same database. However, any data set analyzed with 1.x software must be re-analyzed from scratch (based on the unprocessed images produced by mask application) using the 2.x version of the software. In the same way, any data sets analyzed with 2.x must be re-analyzed from scratch to be analyzed with the 1.x software.

Running under Windows, version 2.0 of the image analysis software does not require any OS upgrades. Running under Linux, version 2.0 requires scientific linux 7.0. The upgrade to linux 7.0 from 6.0 generally necessitates a complete reinstallation of linux OS and re-installation image server software.

Versions 1.x and 2.x image analysis software can use the same database. **However, both versions cannot be run *at the same time* on the same database**. You should run *either* 1.x *or* 2.x image analysis servers. To analyze a data set with v2.0, shut down all 1.x analysis nodes and run 2.x nodes. Any scheduled jobs will run using the 2.x software. For any data analyzed using 2.x software, you must use a 2.x version of the worm browser to inspect that analysis.
To go back to 1.x analysis, shut down all 2.x analysis nodes, schedule new analysis jobs to re-analyze a data set, and use 1.x version of the worm browser.

Upgrade instructions

1. **Backup your mysql database.** On a linux machine (for example one of the image acquisition machines) open terminal window, type the command
   mysqldump –u root –p image_server | gzip > mybackup.sql.gz
   If you have a large database, this command may take a long time, in some cases hours.
2. **Update the analysis software on image analysis server.**
   On Windows: Download the latest image server and worm browser executables from the github repository, branch flow .
   On Linux: If you are not running linux 7, you will need to install it. This will generally require that you wipe the computer and start from scratch, following the directions in this document.   This can be done by following the instructions in this document "Linux: Installing Scientific Linux on the Image Acquisition PC", and then following the instructions in this document "Linux: Installing the image server software".
3. **Update the central sql database schema to support v2.0**
   On any machine, open a terminal window (linux) or a "Command Prompt" (windows). Navigate to the directory in which the executables are stored (on linux, this would be *cd linux*

/usr/local/bin/ns_image_server).

Type the command *./ns_image_server update_sql* (on windows you may omit the leading ./ characters).

4. **Update the sql schema of each acquisition server**
   This step must be performed on each machine running an image acquisition server. Each image acquisition server has a local sql db used for caching temporary information, and the each of these sql dbs must be updated.
   If you have upgraded the image acquisition server software on this machine to software v2.x (upgrading to scientific linux 7, downloading the new source code and compiling it) type the command
   ns_image_server update_sql
   The local schema will be updated automatically.
   *If you have **not** upgraded the image acquisition server software on this machine to software v2.x*:
   a. Open a terminal window and type
      mysql -u root -p image_server_buffer
      You will need to enter the root sql password you set for this computer. This will open a mysql prompt
   b. Type / copy and paste each of the following two commands, typing enter after each one.
      ALTER TABLE buffered_host_event_log` ADD COLUMN `sub_text` MEDIUMTEXT NOT NULL AFTER `processing_duration`, ADD COLUMN `node_id` INT NOT NULL DEFAULT '0' AFTER `sub_text`;

      ALTER TABLE `buffered_experiments` ADD COLUMN `mask_time` BIGINT(20) UNSIGNED NOT NULL DEFAULT '0' AFTER `number_of_regions_in_latest_storyboard_build`, ADD COLUMN `compression_type` CHAR(50) NOT NULL DEFAULT 'jp2k' AFTER `mask_time`, ADD COLUMN `compression_ratio` FLOAT NOT NULL DEFAULT '0.05' AFTER `compression_type`;

   Be sure to repeat this steps on *every* machine running an image acquisition server.

5. **Update PHP scripts running the lifespan machine web interface.** You may want to keep a copy of the old scripts, so both the v1.0 and v2.0 versions can be used.
   a. On the linux machine running the scripts, move the old scripts to a new folder by opening a terminal window and typing the command
      mkdir /var/www/html/v1
      mv /var/www/html/index.php /var/www/html/v1
      mv –r /var/www/html/image_server_web var/www/html/v1
   b. In this way, the old scripts will be still accessible through the web browser at the new address http://myserver/v1/ where mysever is the previous url you used to access the web interface
   c. Now follow the instructions in the section of this document entitled "Linux: Install and Configure the image server web interface". This will install and configure the new scripts

Upgrading from 1.8 to 1.9

Important: This upgrade should not be performed while any experiments are running. The upgrade of the SANE drivers requires a change to image capture parameters that is difficult to alter in schedules that are already running.

Linux: Updating the image capture server

1. Update to the latest version of scientific linux 6 (currently 6.8)
   1. *yum update*
   2. *yum install yum-conf-sl6x*
   3. *yum update*
   4. *(again)*
   5. *yum update*
   6. *(will require a restart)*
2. Install openjpeg version 2.1
   1. *cd external_compile_libraries*
   2. *cd openjpeg-build-2.1.1/*
   3. *mkdir build*
   4. *cmake ../*
   5. *make install*
3. ~~Install ITK (this will likely take a long time > 1h)~~
   1. *~~Go to a directory outside the lifespan git repository (e.g. not a subdirectory of lifespan\ )~~*
   2. *~~git clone git://itk.org/ITK.git~~*
   3. *~~cd ITK~~*
   4. *~~mkdir build~~*
   5. *~~cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_CXX_FLAGS:STRING=-fkeep-inline-functions ../~~*
4. install xvid version 1.3.3
   1. *cd xvidcore-1.3.4/build/generic*
   2. *sh bootstrap.sh*
   3. *chmod 755 ./configure./*
   4. *./configure*
   5. *rm /usr/local/lib/libxvidcore.**
   6. *make install*

On the windows machine running the worm browser

1. Download the latest ns_image_server and ns_worm_browser binaries from github repository, directory binaries\windows
2. From the same directory on github, obtain the three dll files xvidcore.dll, openjp2.dll, and libmysql.dll, and place them in the same directory as the windows binaries.
3. Run the worm browser
4. Select the menu option Config/Update Database Schema . This will upgrade your database schema to the latest version. No data will be changed.

~~(optional)Compiling ITK on windows~~

~~In cmake, enable the options ITK_USE_GPU ITK_USE_FFTWF ITK_USE_FFTWD ITK_USE_SYSTEM_FFW Change all the /MD flags to /MT to avoid using debug C++ standard libraries. Disable BUILD_TESTING~~

Upgrading from 1.7 to 1.8
On the linux image capture server

Important: This upgrade should not be performed while any experiments are running. The upgrade of the SANE drivers requires a change to image capture parameters that is difficult to alter in schedules that are already running.

1. Uninstall any older version of SANE
    a. Enter the directory containing the old SANE source code, usually located in /home/image_server/lifespan/external_compile_libraries/sane-backends-1.0.22
    b. Run the command *./make uninstall*
2. Uninstall any older version of libdmtx
    a. Enter the old libdmtx source directory, usually located in /lifespan/external_compile_libraries/libdmtx-0.5.2.
    b. Run the command *./make uninstall*
3. Obtain the latest version of the source code from the github.com repository (see the previous section regarding downloading this repository)
4. Unzip external libraries and external_compile_libraries archives:
    a. Enter the base directory of the lifespan machine source code /home/image_server/lifespan/
    b. Run the command *tar -xvjf external_compile_libraries.tar.bz2*
5. Install new version of SANE
    a. Enter the new SANE source directory, usually located in /home/image_server/lifespan/external_compile_libraries/sane-backends-1.0.24
    b. Run the command *./make install*
6. ***Very important!*** Update all image capture schedules you plan to run to use the new TPU8x10 scan option. New capture schedules should read
   <default_capture_configuration_parameters>--mode=Gray --format=tiff --source="TPU8X10" --depth=16 </default_capture_configuration_parameters>
7. Install new version of libdmtx
    a. Enter the new libdmtx source directory, usually located in /home/image_server/lifespan/external_compile_libraries/libdmtx-0.7.4
    b. Run the commands
       sh autogen.sh
       ./make install
8. Update the web interface
    a. Run the script *sh /home/image_server/lifespan/ns_install_website.sh*
    b. In emacs, open the file /var/www/html/image_server_web/ns_image_server_website.ini and add the line
       $default_database = "image_server";
    c. This line can be added anywhere in the file.

On the windows machine running the worm browser

5. Obtain the latest ns_image_server and ns_worm_browser binaries from github
6. Run the worm browser

7. Select the menu option Config/Update Database Schema . This will upgrade your database schema to the latest version. No data will be changed.

Appendix 1: Useful Linux Commands
(as suggested by Deborah McEwan and Annie Conery)

cd   changes directory (cd .. goes up one level)

pwd   lists current directory

ls   lists files in folder

ls –all   see all files including hidden ones. If a file or directory has a period (.) as the first character in its name, it is by default not listed by the *ls* command.   For example, *ls* will not list the file .cifspass , but *ls –all* will.

mkdir   makes a directory

emacs   document editing

vi   document editing

su   log into the user root account (required, for example, to restart the computer).

Ifconfig   get the IP address of the current machine

<control>c   force quit