Server Operating Systems

Lecture 11
Compiling Software

Compilation

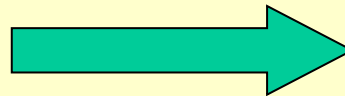This is the process of taking a program written in

     SOURCE CODE   (C++, Basic, Java)

and converting it into

     OBJECT CODE   (Binary machine code)

A program has to be in machine code before it can be executed.

```
main()
{
let x=4*6;
write( x );
}
```

```
10101100
01110010
10101010
11000011
```

The trouble is, each type of processor has it's own unique set of machine code instructions.

Why compile?

Normally, when you install software on a windows system, it is already compiled.

All you can get are the executables. The .exe and .dll files.

PROBLEM:
The software companies don't know what sort of processor you have.

That is why you must be careful to get the correct version of the software. You can get executables that have been compiled to run on the following architectures:

- 32 bit x86
- 64 bit x86
- 32 bit PowerPC
- 64 bit PowerPC
- 64 bit Itanium
- or others

Even then, the software manufacturers do not know the details of your hardware.

Benefits of Compiling Software

They provide executables compiled to run on the lowest common denominator hardware spec.

You may have the most advanced processor, and best graphics card there is.

BUT:

If your software hasn't been compiled for that specific hardware, it will not run as quickly or as smoothly as it could.

When you compile software, the compiler will examine your hardware, and generate machine code which is customised for your specific machine.

To do this, you need the source code.

Why don't Microsoft (and others) give you access to their source code?

Linux Packages

Linux distributions also allow you to install pre-compiled binary packages.

There are several methods.

rpm   Red Hat package manager – used by Fedora, Suse, Mandriva

deb   Debian packages – used by Debian, Ubuntu

An rpm or deb package is actually a collection of related files and programs which can be installed as one unit.

On Suse, Yast2 > Software Management maintains a database of all RPMs that have been installed, and when you are installing a new one, it will check for **dependencies** and install them as well.
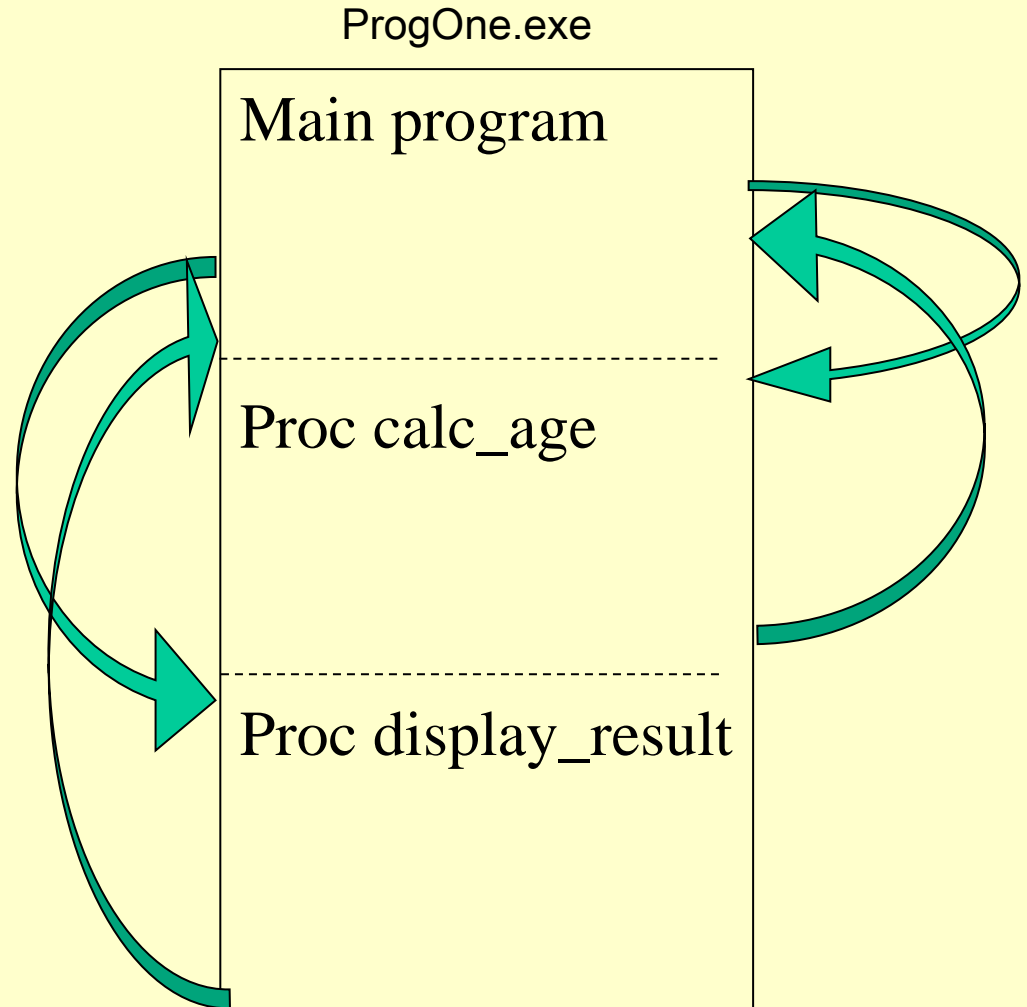
Makes life a lot easier.

YAST2 Package Manager
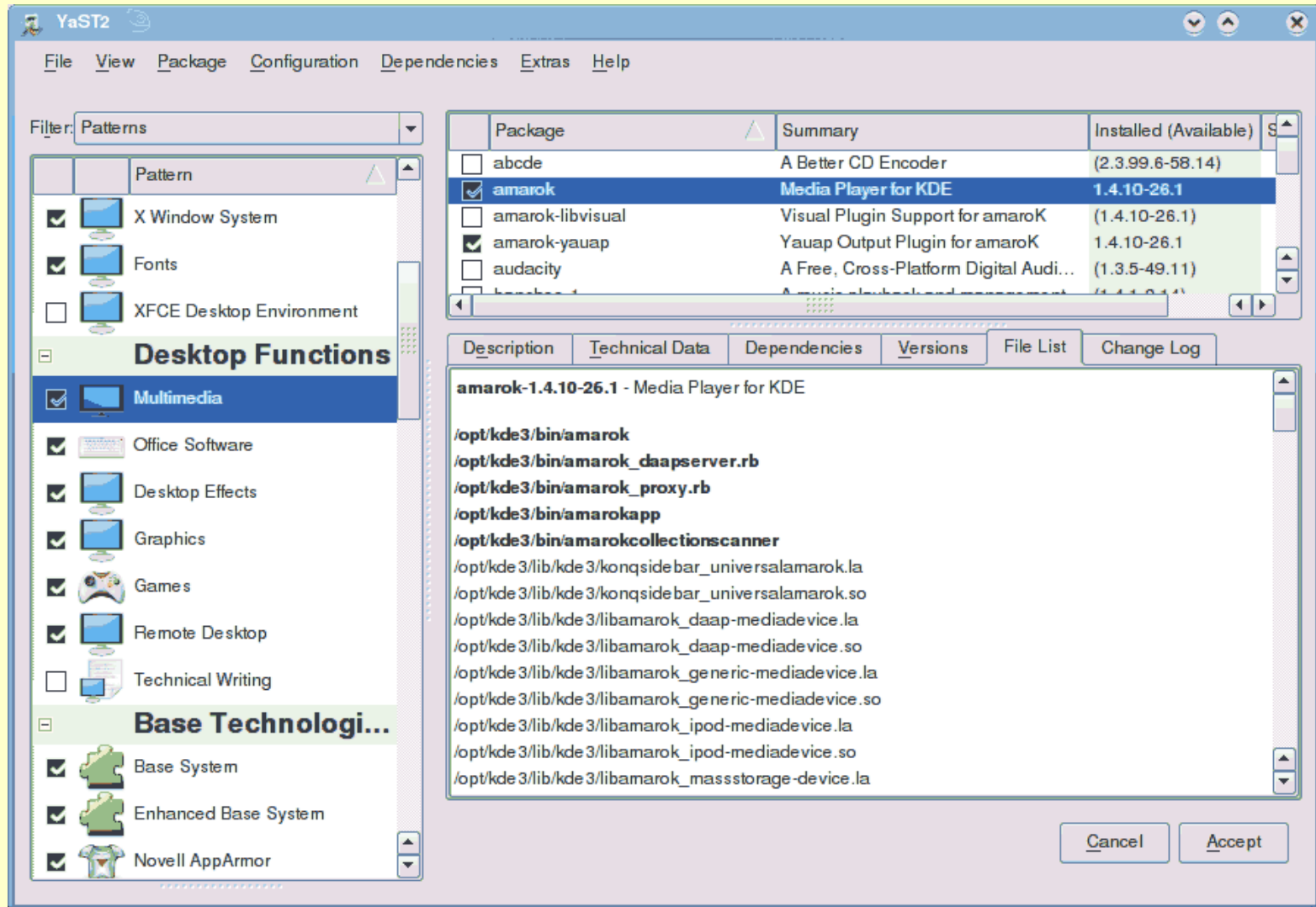
What are Dependencies?

In a traditional program, the code is divided up into procedures, or functions.

These procedures can be called from the main part of the program.

ProgOne.exe

Main program

Proc calc_age

Proc display_result

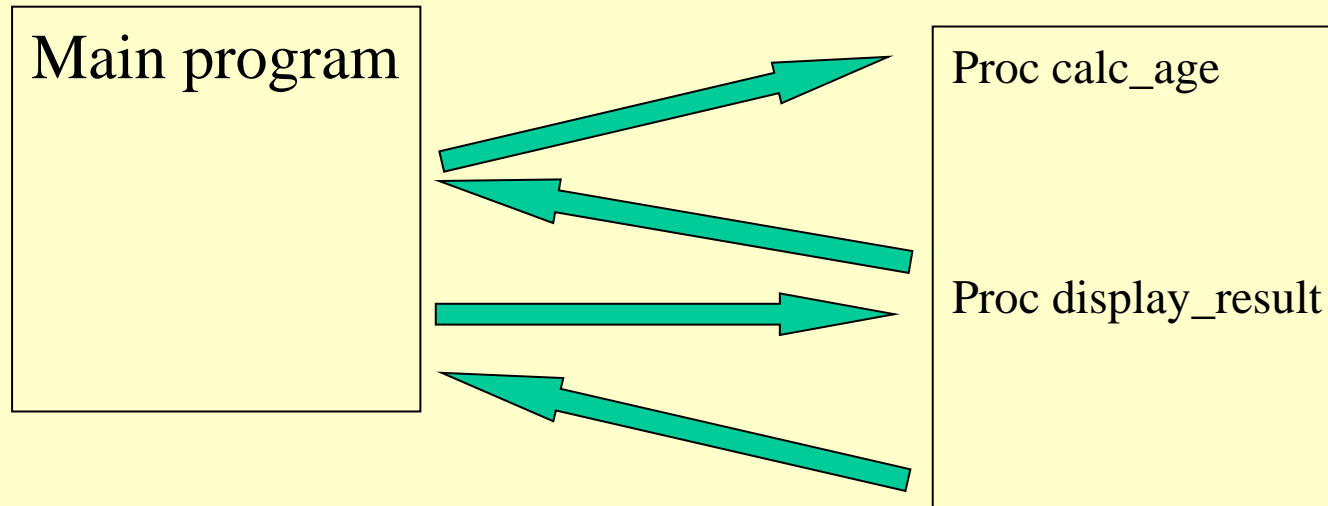# Libraries – Dependencies

The main part of the program is usually packaged separately from the functions.

ProgOne.exe

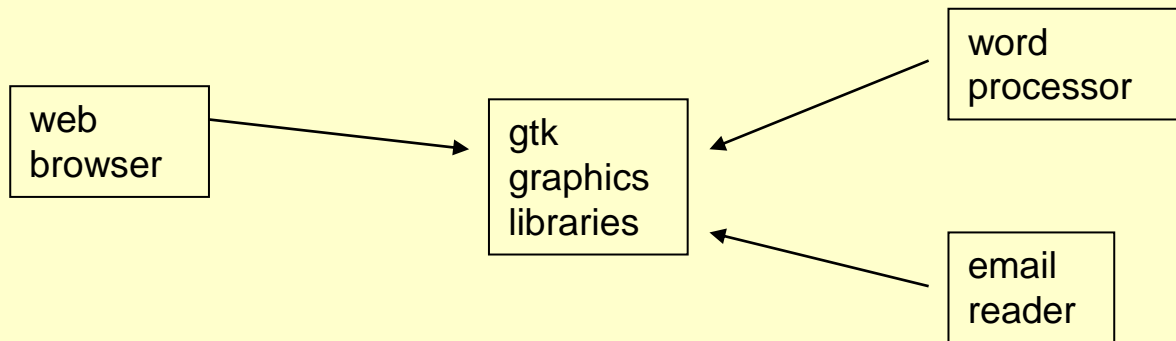library.dll

| Main program |

| Proc calc_age |

| Proc display_result |

What happens if I install ProgOne.exe but not library.dll ?

Linux does not use the .exe and .dll file extensions, but the same principles apply.

The /lib folder contains a lot of standard library packages, which can be used by different executables.

```
┌──────────┐                      ┌──────────┐      ┌──────────┐
│  web     │                      │  gtk     │ ←──  │  word    │
│  browser │ ──────────────────→  │  graphics│      │ processor│
└──────────┘                      │  libraries│     └──────────┘
                                  │          │ ←──  ┌──────────┐
                                  └──────────┘      │  email   │
                                                    │  reader  │
                                                    └──────────┘
```

BUT: This can lead to problems:

program1   **compiled against**   **gtk library v1.2.3**

program2   **compiled against**   **gtk library v2.0.2**

Installing individual libraries is usually more trouble than it is worth.

Much easier to let the distributions (Suse, Fedora, Debian) do all that for you.

Linux software

You can download and install pre-compiled binary RPM packages using YAST2 on SUSE.

–   Security implications?
–   What if no one has packaged up the software that you want, as an RPM?

You can download the source code and compile it yourself.

There are many sites which make source code available:

–   The programmers own website.
–   Well known independent repositories.   Sourceforge, Freshmeat
–   Linux distribution repositories.   Opensuse, Packman

Browse galculator Files on SourceFor...

source forge FIND AND DEVELOP OPEN SOURCE SOFTWARE

Find Software | Develop | Create Project | Blog | Site Support | About

SourceForge.net > Find Software > galculator > Browse Files

galculator by chimaira

Summary | Files | Support | Develop

galculator is a GTK 2 based scientific calculator supporting algebraic mode as well as RPN. Features include arithmetic precedence handling, mathematical functions, different number bases (DEC, HEX, BIN, OCT) and angle bases (radiant, degree, and grad).

Download Now!
galculator-1.3.4.zip (599.9 KB)   OR   View all files ▸

Browse Files for galculator

File/Folder Name	Platform	Size	Date ↓	Downloads	Notes/Subscribe
Newest Files					
galculator-1.3.4.zip		599.9 KB	2009-02-08	637	
galculator-1.3.4.tar.gz		532.5 KB	2009-02-08	1,146	
galculator-1.3.4.tar.bz2		393.8 KB	2009-02-08	2,665	
All Files					
▼ galculator		13.3 MB	2009-02-08	24,837	
▶ 1.3.4		1.5 MB	2009-02-08	4,448	
▶ 1.3.3		1.5 MB	2009-01-11	818	
▶ 1.3.2		1.5 MB	2009-01-06	407	

Done

galculator is a GTK 2 based calculator program.

Available in three different archive formats.

zip

tar.gz

tar.bz2

The latter two are called 'tarballs'.

The convention is to include the version number in the file name.

Checking the tarball

Some websites that host software packages also make available an md5 checksum.

This is a number that has been generated by performing a complex calculation on the file containing the software package.

You perform the same calculation on the tarball once you have downloaded it.

If the numbers match, you know that the tarball has not been corrupted or altered.

Checking the tarball

For example:

Download   **bluefish-1.0.7.tar.gz**   the software

Download   **bluefish-1.0.7.tar.gz.md5**   text file

Run the following command   **md5sum bluefish-1.0.7.tar.gz**

which will output the checksum: **9e107d9d372bb6826bd81d3542a419d6**

if the text file also contains: **9e107d9d372bb6826bd81d3542a419d6**

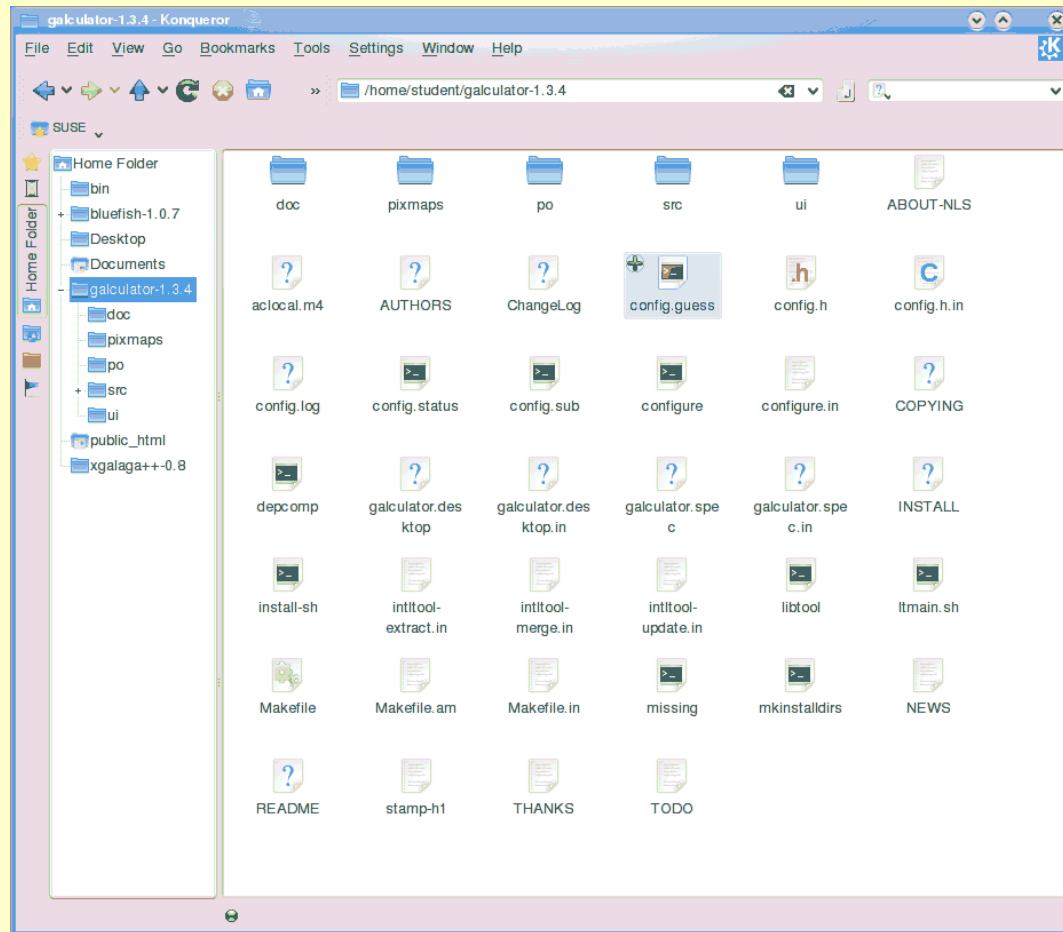then the file has not been corrupted or altered.
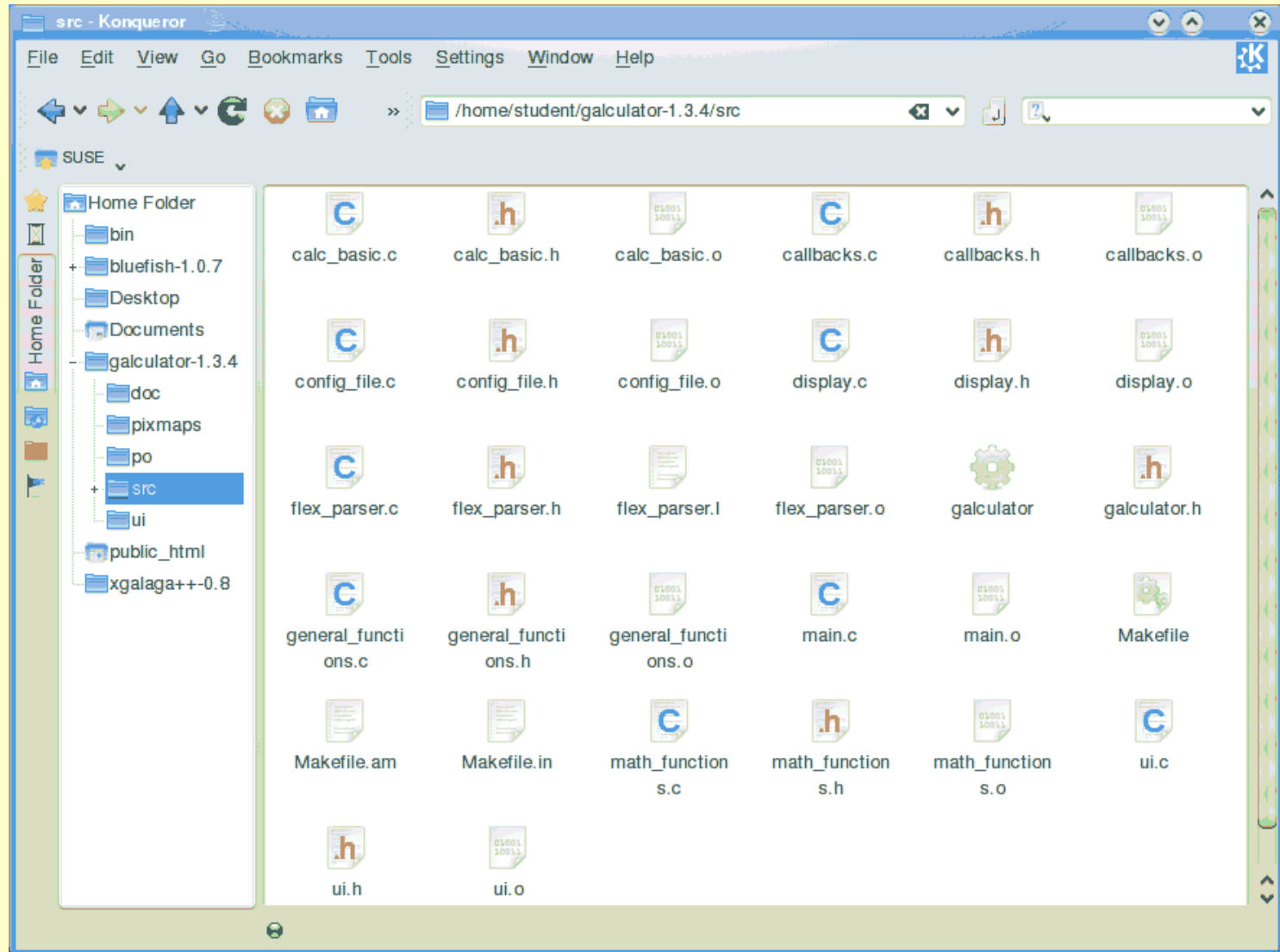
Unpack the tarball

GUI   - right click on the file, and extract.

Command Line   – Use the command: `tar xzpf galculator-1.3.4.tar.gz`

Source Code Files

Contents of the tarball

This will usually create a folder with the same name, containing:

– all of the source code files needed.

– installation scripts

– the following text files:

- README   General info about the software
- INSTALL   Installation instructions, inc details of any libraries required
- TODO   Future improvements to be made to the next version
- AUTHORS   Who wrote it, with contact details
- COPYING   Terms of licence – Usually the GPL or some variant.
- NEWS   Changelist – differences between the different versions

Usually worth checking INSTALL as the following method isn't always followed.

Compilation is always done from the CLI

cd into the tarball folder: `cd galculator-1.3.4`

make sure all files are executable: `chmod –R 755 *`

Examine hardware and create makefile: `./configure`

Read error messages, if any, and install required library packages using YAST2

Compile the software: `make`

Change to root user: `su`

Install the software: `make install`

Return to user: `exit`

Run software: `galculator`
                                                                              `(or whatever the program is).`

Preparation

WARNING:

Before we do any of this on OpenSuse for the first time,
we need to do the following:

Switch off autorefresh in the package manager:

- – This will stop YAST2 trying to log on to the internet to do automatic software updates.

Install the compilers and standard libraries:

- – OpenSuse doesn't install the software development packages by default.

Start > System > Yast2 > Software > Software Repositories
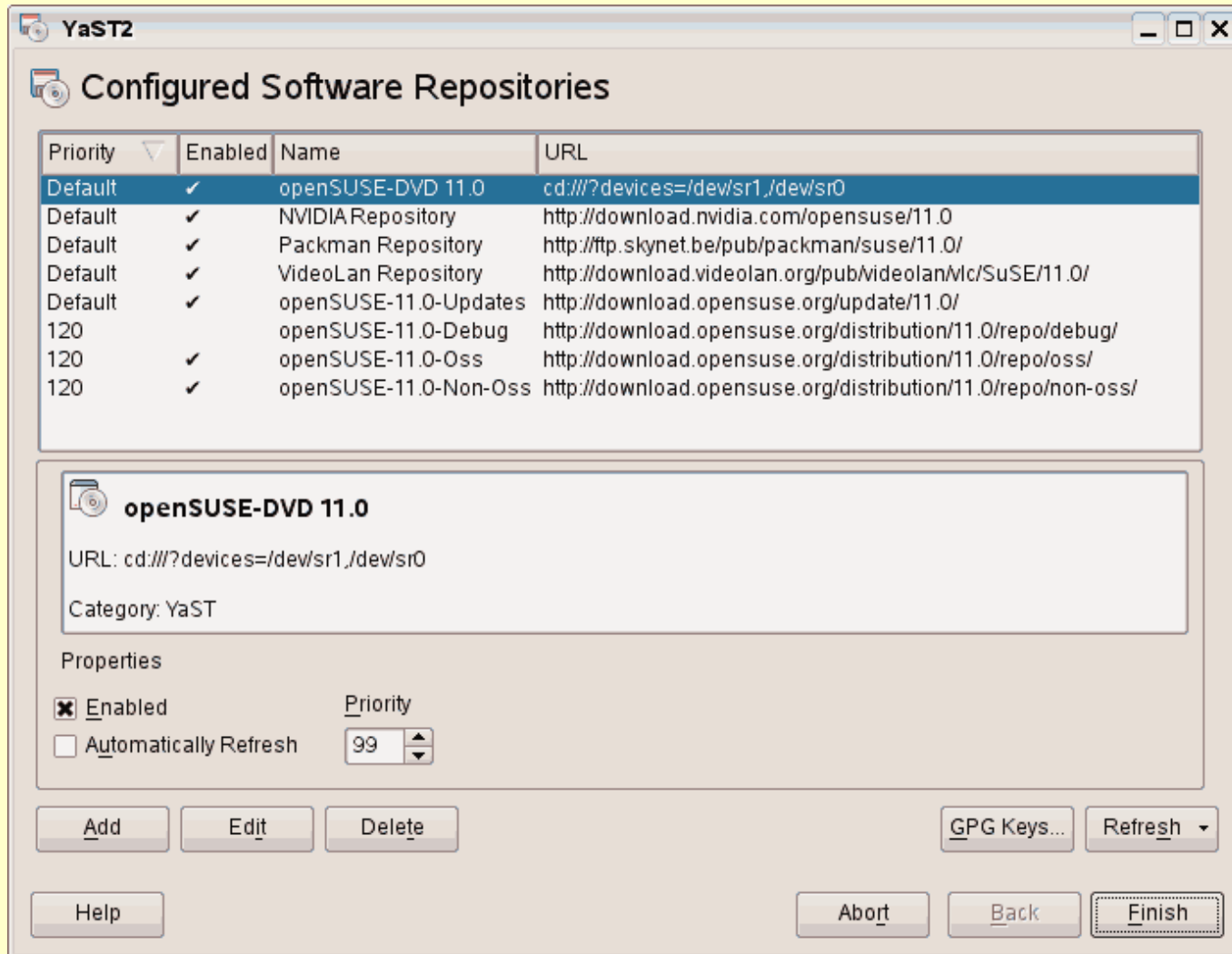
Start > System > YAST2 > Software > Software Management