ECOL 553L

Introduction to Formats, UNIX, and VIM

File Formats and Introduction to Unix

- “Today”s Topics:
  - Sequence File Formats
  - Unix Intro
    - Basic Commands
    - SSH
  - VIM (file editing)

A note about grades

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens
- Lab Grades:

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens
- Lab Grades:
  - Homework 40%

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens
- Lab Grades:
    - Homework 40%
    - Quizzes 50%

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens
- Lab Grades:
  - Homework 40%
  - Quizzes 50%
  - Class Participation 10%

A note about grades

- The Syllabus has been on the (now down) server, is now temporarily at http://www.cs.arizona.edu/~deblasio/ECOL553/
- You will find the tentative semester schedule, as well as the grading policy

- Readings are from “Beginning Perl” by Simon Cozens
- Lab Grades:
  - Homework 40%
  - Quizzes 50%
  - Class Participation 10%
  - (lowest quiz grade and lowest homework grade dropped)

A note about grades

A note about grades

- There will be ~1 homework and one quiz per week

A note about grades

- There will be ~1 homework and one quiz per week
  - Probably on Tuesdays

A note about grades

- There will be ~1 homework and one quiz per week
  - Probably on Tuesdays
- The in class quizzes are timed but open book/computer

A note about grades

- There will be ~1 homework and one quiz per week
  - Probably on Tuesdays
- The in class quizzes are timed but open book/computer
- The homework will be assigned at the begining of the week (on Tuesday) and will be due one week after

A note about grades

- There will be ~1 homework and one quiz per week
  - Probably on Tuesdays
- The in class quizzes are timed but open book/computer
- The homework will be assigned at the begining of the week (on Tuesday) and will be due one week after
  - This week is an exception

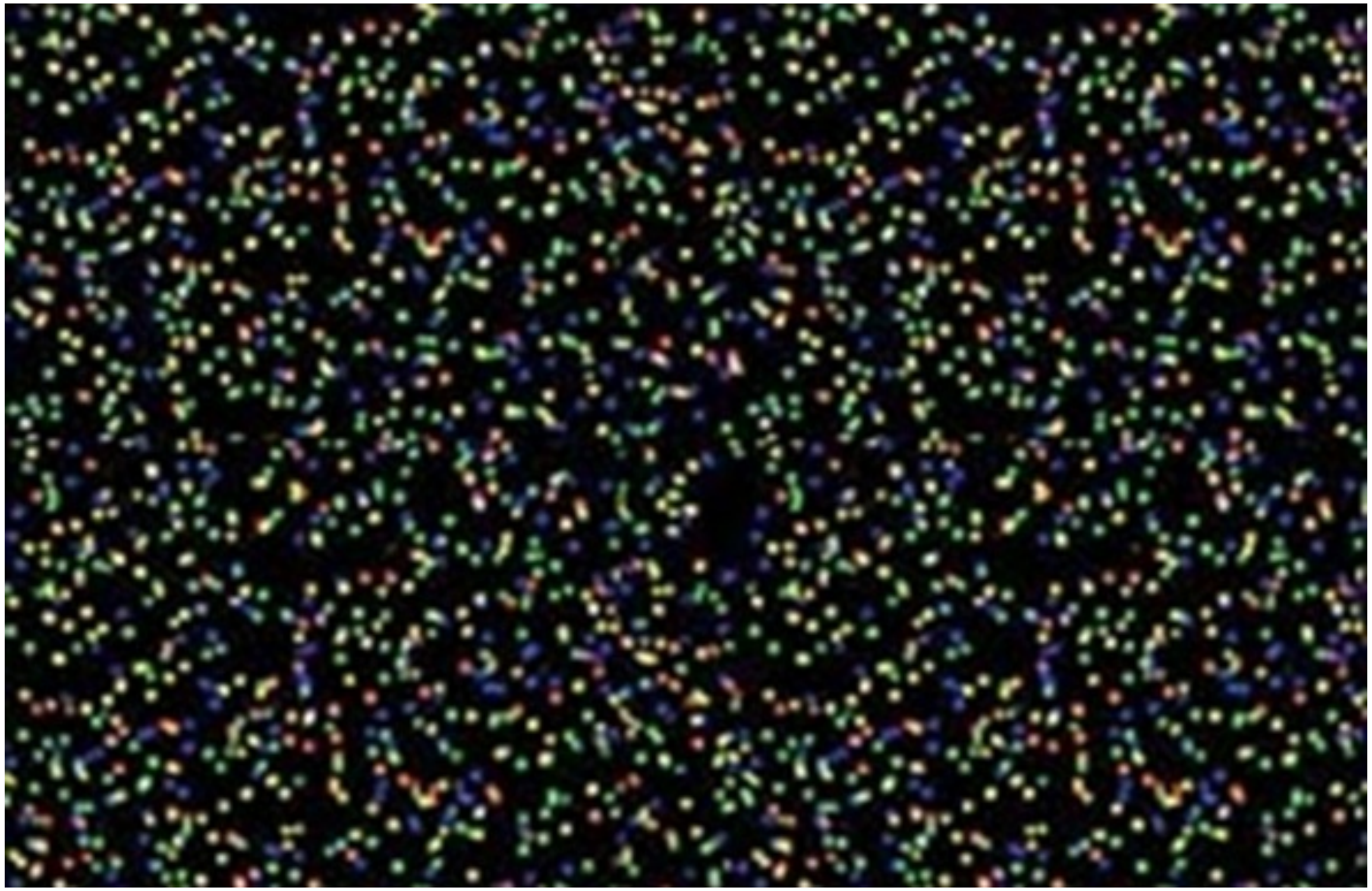
The Taxonomy of (Biological) File Formats

The Taxonomy of (Biological) File Formats

- Raw Data

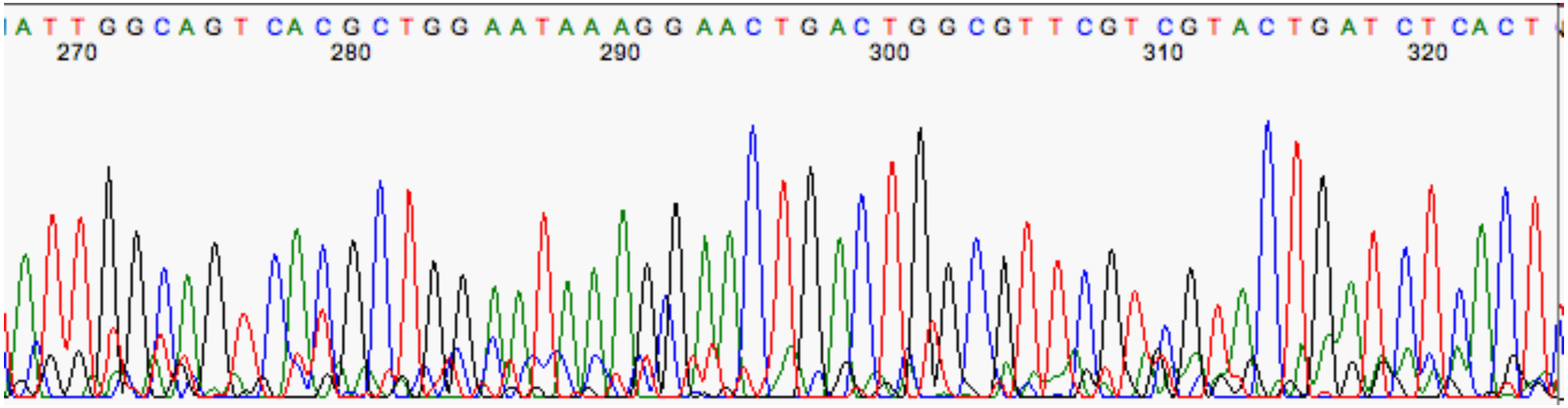
The Taxonomy of (Biological) File Formats

- Raw Data
  - sequencer images (think back to what Dr. Barker was talking about last week)



The Taxonomy of (Biological) File Formats

- Raw Data
  - sequencer images (think back to what Dr. Barker was talking about last week)
  - chromatograph data



The Taxonomy of (Biological) File Formats

- Raw Data
  - sequencer images (think back to what Dr. Barker was talking about last week)
  - chromatograph data
- Sequence Data

The Taxonomy of (Biological) File Formats

- Raw Data

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

- Sequence Data

The Taxonomy of (Biological) File Formats

- Raw Data

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

- Sequence Data
  - 3 major components

The Taxonomy of (Biological) File Formats

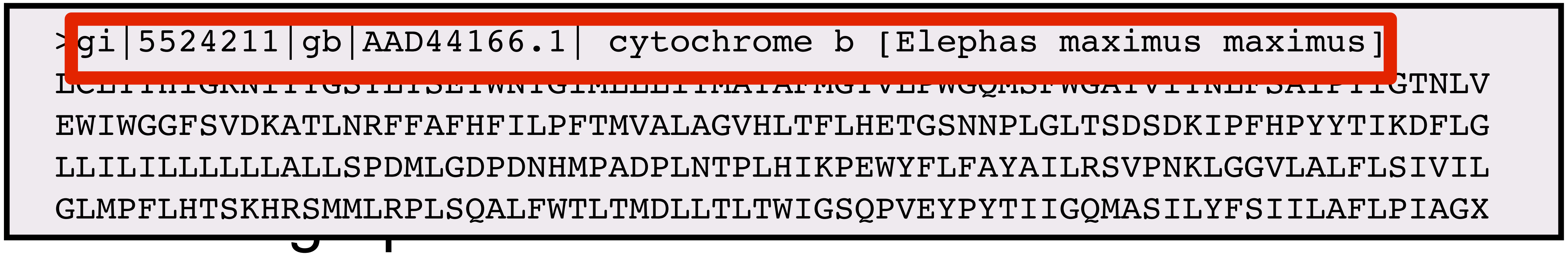
- Raw Data

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

- Sequence Data
  - 3 major components
    - sequence (nucleotide, amino acid, structure, quality...)

The Taxonomy of (Biological) File Formats

- Raw Data



```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLIIHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

- Sequence Data
  - 3 major components
    - sequence (nucleotide, amino acid, structure, quality...)
    - label (identifier, name)

The Taxonomy of (Biological) File Formats

- Raw Data

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

- Sequence Data
  - 3 major components
    - sequence (nucleotide, amino acid, structure, quality...)
    - label (identifier, name)
    - meta-data [optional] (location, publication, species...)

The Taxonomy of (Biological) File Formats

The Taxonomy of (Biological) File Formats

- Sequence Data

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
```

Sometimes programs trim the name of sequences! Be Careful!

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal

```
CLUSTAL W (1.82) multiple sequence alignment


FOSB_MOUSE      MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60
FOSB_HUMAN      MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA 60
                ************************************************************

FOSB_MOUSE      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGASGS 120
FOSB_HUMAN      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPVVDPYDMPGTSYSTPGMSGYSSGGASGS 120
                ********************************.***************:*.**:******

FOSB_MOUSE      GGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRRVRRERNKLAAAKCRNRRRELT 180
FOSB_HUMAN      GGPSTSGTTSGPGPARPARARPRRPREETLTPEEEEKRRVRRERNKLAAAKCRNRRRELT 180
                ****** ***** .**********************************************

FOSB_MOUSE      DRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGCKIPYEEGPGPGPLAEVRD 240
FOSB_HUMAN      DRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGCKIPYEEGPGPGPLAEVRD 240
                ************************************************************

FOSB_MOUSE      LPGSTSAKEDGFGWLLPPPPPPPLPFQSSRDAPPNLTASLFTHSEVQVLGDPFPVVSPSY 300
FOSB_HUMAN      LPGSAPAKEDGFSWLLPPPPPPPLPFQTSQDAPPNLTASLFTHSEVQVLGDPFPVVNPSY 300
                ****:.******.**************:*:**************************.***

FOSB_MOUSE      TSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPSLLAL 338
FOSB_HUMAN      TSSFVLTCPEVSAFAGAQRTSGSDQPSDPLNSPSLLAL 338
                ***********************:**************
```

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal
  - GenBank

```
LOCUS       AB000263                368 bp    mRNA    linear   PRI 05-
FEB-1999
DEFINITION  Homo sapiens mRNA for prepro cortistatin like peptide, complete
            cds.
ACCESSION   AB000263
ORIGIN
        1 acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg
       61 ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
      121 caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc
      181 aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
      241 gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga
      301 agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca
      361 gacctgaa
//
```

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal
  - GenBank
  - SAM

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
r001 163 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5H6M * 0 0 AGCTAA * NM:i:1
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 16 ref 29 30 6H5M * 0 0 TAGGC * NM:i:0
r001 83 ref 37 30 9M = 7 -39 CAGCGCCAT *
```

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal
  - GenBank
  - SAM
  - Variants

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal
  - GenBank
  - SAM
  - Variants
    - FASTQ

```
@NCYC361-11a03.q1k bases 1 to 1576
GCGTGCCCGAAAAAATGCTTTTGGAGCCGCGCGTGAAAT...
+NCYC361-11a03.q1k bases 1 to 1576
!)))))****(((***%%((((*(((+,**(((+***+,-...
```

The Taxonomy of (Biological) File Formats

- Sequence Data
  - Fasta
  - Clustal
  - GenBank
  - SAM
  - Variants
    - FASTQ
    - Anootated Fasta

```
# STOCKHOLM 1.0

AL929536.7/77590-77911     GGAUGUGAGGGCGAUCUGGCUGCGACAUCUGUCACCCCAUUGAUCGCUAG
M63671.1/163-492           GGAUGUGAGGGCGAUCUGGCUGCGACAUCUGUCACCCCAUUGAUCGCCAG
K02909.1/1-330             GGAUGUGAGG.CGAUCUGGCUGCGACAUCUGUCACCCUAUUGAUCGCCAG
X05490.1/245-574           GGAUGUGAGG.CGAUCUGGCUGCGACAUCUGUCACCCCAUUGAUCGCCAG
X04236.1/2-328             .GAUGUGAGGGCGAUCUGGCUGCGACAUCUGUCACCCCAUUGAUCGCCAG
X04992.1/244-574           GGAUGUGAGGGCGAUCUGGCUGCGACAUCUGUCACCCCAUUGAUCGCCAG
#=GC SS_cons               <<<.<<<<<<<.<<...<<....<<<<<<<<<<.<<<<<..<<<<<<<..
```

The Taxonomy of (Biological) File Formats

The Taxonomy of (Biological) File Formats

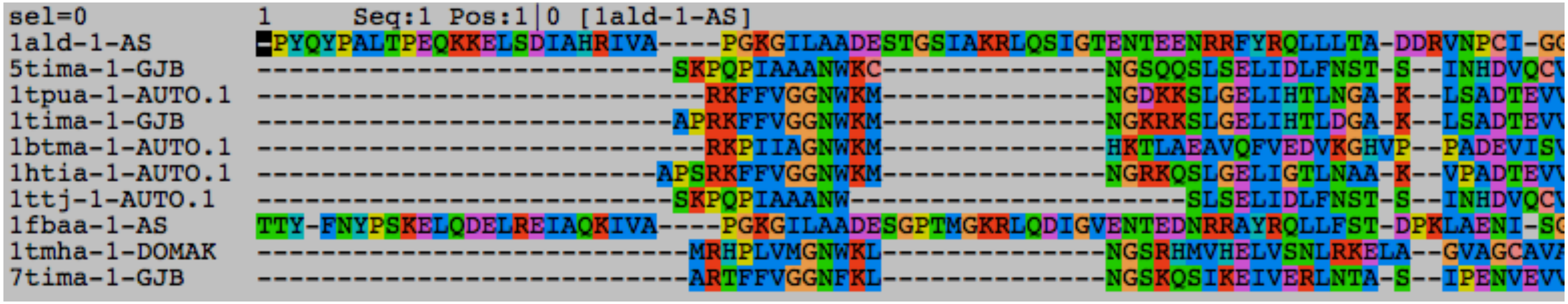
- Viewing

The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments

The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments
    - Seaview



```
sel=0           1     Seq:1 Pos:1|0 [1ald-1-AS]
1ald-1-AS       -PYQYPALTPEQKKELSDIAHRIVA----PGKGILAADESTGSIAKRLQSIGTENTEENRRFYRQLLLTA-DDRVNPCI-GG
5tima-1-GJB     --------------------------SKPQPIAAANWKC--------------NGSQQSLSELIDLFNST-S--INHDVQCV
1tpua-1-AUTO.1  ----------------------------RKFFVGGNWKM--------------NGDKKSLGELIHTLNGA-K--LSADTEVV
1tima-1-GJB     --------------------------APRKFFVGGNWKM--------------NGKRKSLGELIHTLDGA-K--LSADTEVV
1btma-1-AUTO.1  ----------------------------RKPIIAGNWKM--------------HKTLAEAVQFVEDVKGHVP--PADEVISV
1htia-1-AUTO.1  -------------------------APSRKFFVGGNWKM--------------NGRKQSLGELIGTLNAA-K--VPADTEVV
1ttj-1-AUTO.1   --------------------------SKPQPIAAANW---------------------SLSELIDLFNST-S--INHDVQCV
1fbaa-1-AS      TTY-FNYPSKELQDELREIAQKIVA----PGKGILAADESGPTMGKRLQDIGVENTEDNRRAYRQLLFST-DPKLAENI-SG
1tmha-1-DOMAK   ---------------------------MRHPLVMGNWKL--------------NGSRHMVHELVSNLRKELA--GVAGCAVA
7tima-1-GJB     ---------------------------ARTFFVGGNFKL--------------NGSKQSIKEIVERLNTA-S--IPENVEVV
```

The Taxonomy of (Biological) File Formats

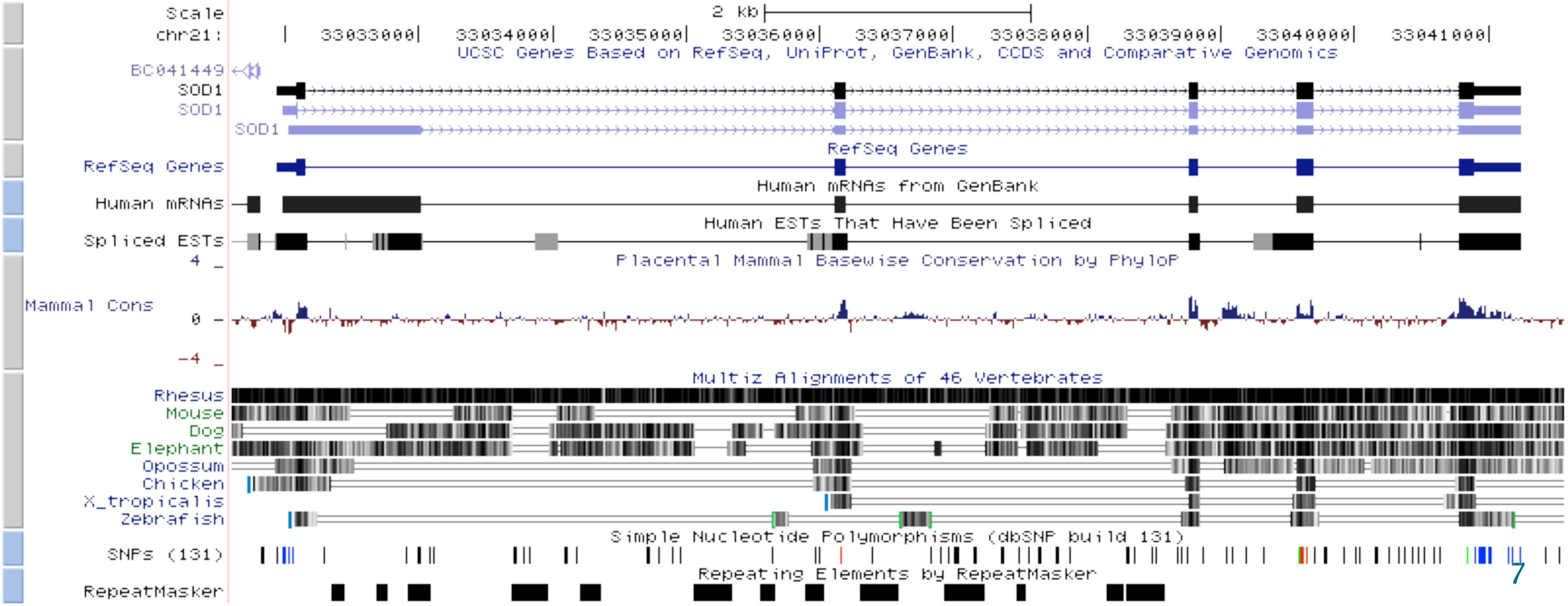
- Viewing
  - Single Alignments
    - Seaview
    - ClustalW

The Taxonomy of (Biological) File Formats

- Viewing
    - Single Alignments
        - Seaview
        - ClustalW
    - Genome Browsers

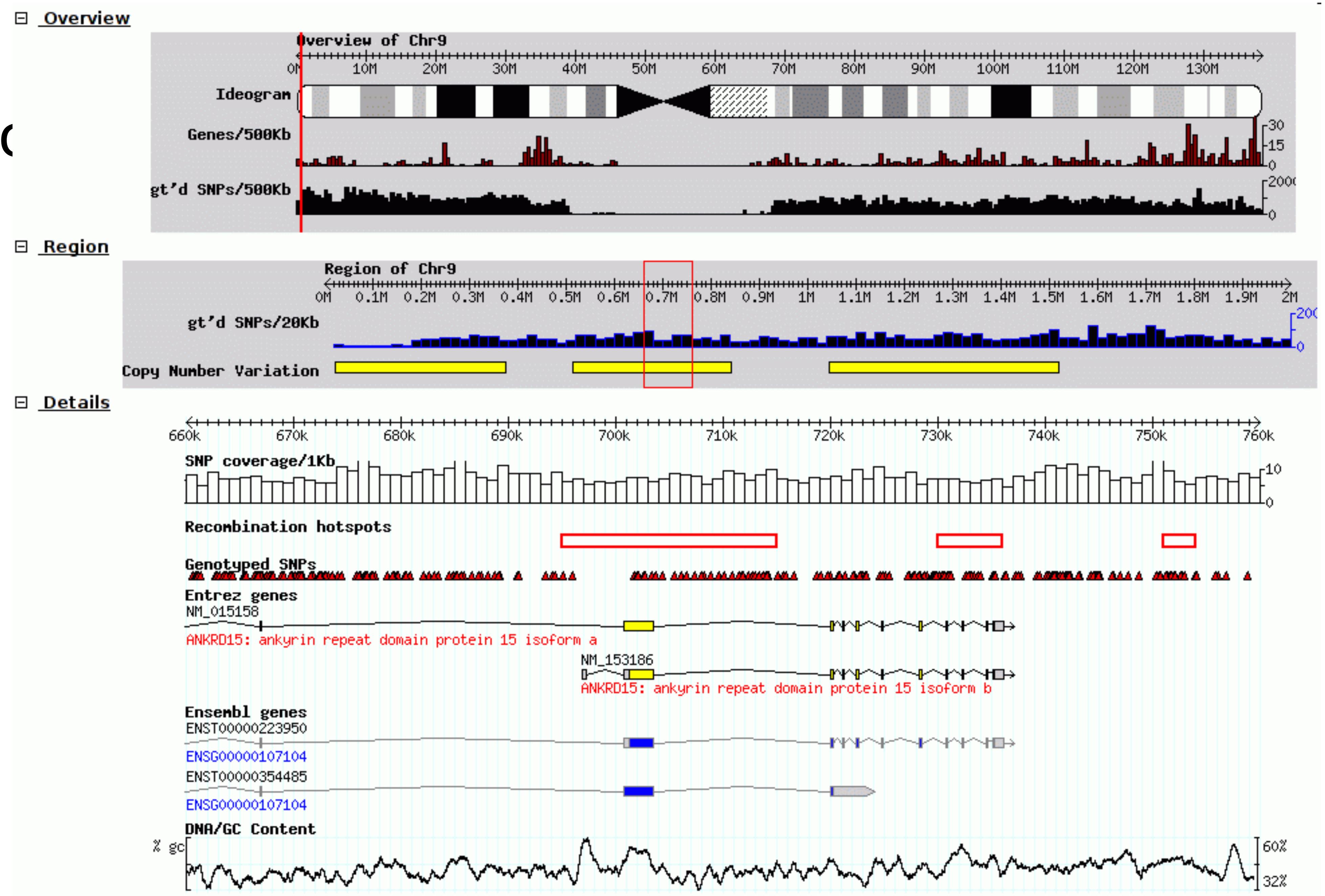
The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments
    - Seaview
    - ClustalW
  - Genome Browsers
    - UCSC



The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments
    - Seaview
    - ClustalW
  - Genome Br
    - UCSC
    - GBrowse



The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments
    - Seaview
    - ClustalW
  - Genome Browsers
    - UCSC
    - GBrowse
    - JBwose

The Taxonomy of (Biological) File Formats

- Viewing
  - Single Alignments
    - Seaview
    - ClustalW
  - Genome Browsers
    - UCSC
    - GBrowse
    - JBwose
    - ...

Phylogeny File Formats

- NEWICK/NHX (phylip)
    - Rooted or unrooted tree representation

      `(A:0.1,B:0.2,(C:0.3,D:0.4):0.5);`
- Nexus (PAUP, MacClade, NDE)
    - Data/descriptor blocks plus analysis commands
    - http://www.paup.csit.fsu.edu/nfiles.html
- phyloXML (forester-atv)
    - Taxonomy, sequence, confidence, events
    - http://www.phyloxml.org/
    - http://sourceforge.net/projects/forester-atv/

- Format converter: http://www.phylogeny.fr/version2_cgi/data_converter.cgi

Converting Sequence File Formats

- http://hcv.lanl.gov/content/sequence/FORMAT_CONVERSION/form.html
  - Online tool for conversion
- http://mybio.wikia.com/wiki/Sequence_format_conversion
  - Links to conversion tools
- BioPerl (we'll use this later), BioPython, BioRuby, BioJava, …
- Galaxy (Penn State)
  - Tools, history, basic workflow, data libraries, sharing
  - Wiki and screencasts
  - http://galaxy.psu.edu/

Useful Links for Learning about UNIX

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)

- BCF Unix Tutorials: http://bcf.arl.arizona.edu

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)

- BCF Unix Tutorials: http://bcf.arl.arizona.edu
  - Follow the 'Site Navigation' and 'Resources' links

Useful Links for Learning about UNIX

- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html
- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)
- BCF Unix Tutorials: http://bcf.arl.arizona.edu
  - Follow the 'Site Navigation' and 'Resources' links

Useful Links for Learning about UNIX

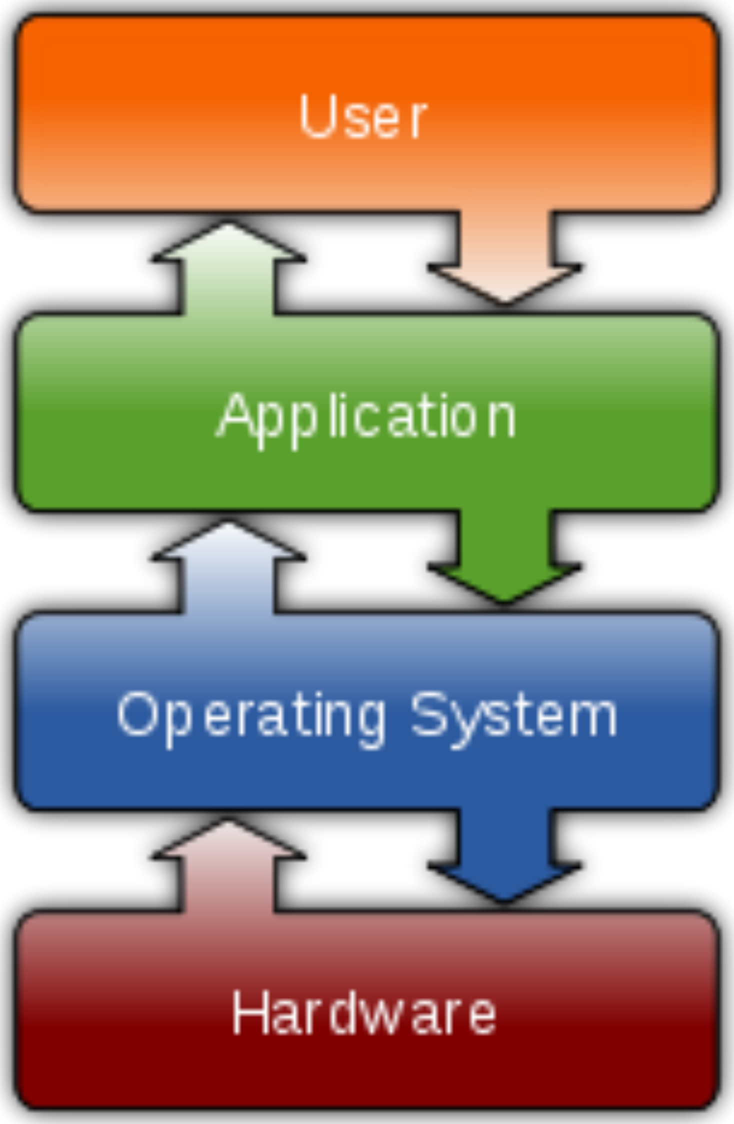
- There are many sources of UNIX tutorials.
  - A few of these are:
    - http://www.ee.surrey.ac.uk/Teaching/Unix/
    - http://www.math.utah.edu/lab/unix/unix-tutorial.html
    - http://www.unixtools.com/tutorials.html

- There are some short Quicktime movies accessible from the UA Computer-Based Training Site (type UNIX in the 'Find a Course' box): http://uacbt.arizona.edu/default.htm
  - (you do not need to purchase these!)

- BCF Unix Tutorials: http://bcf.arl.arizona.edu
  - Follow the 'Site Navigation' and 'Resources' links

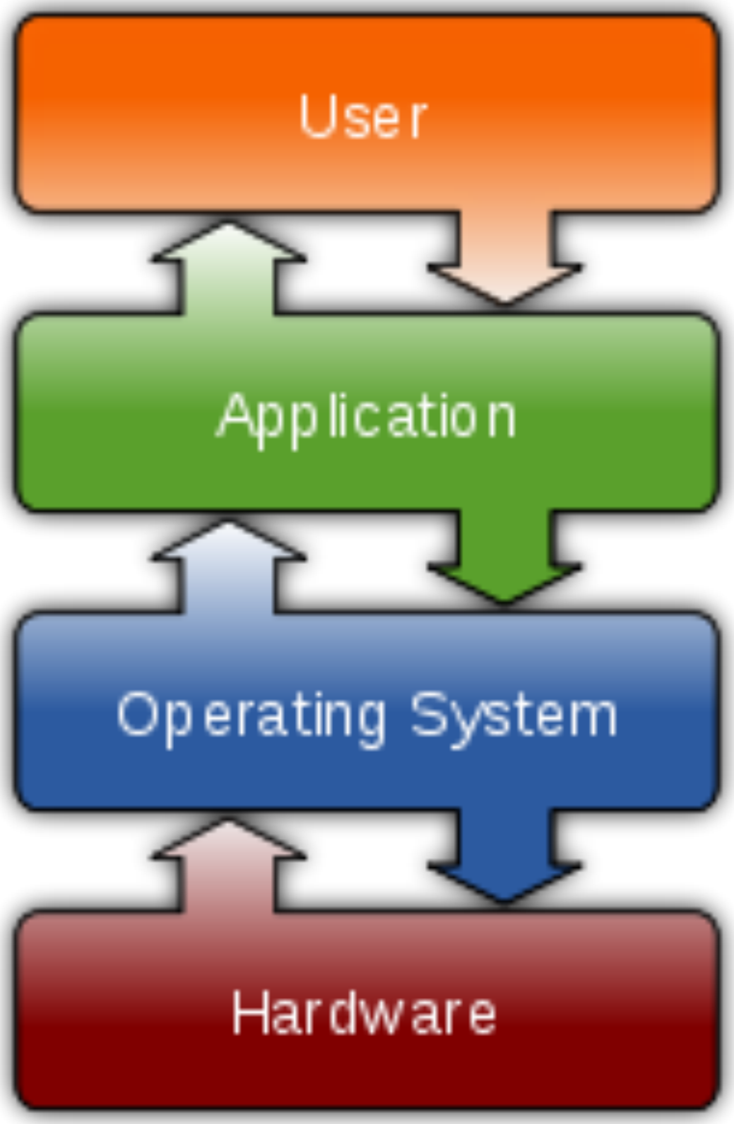
- Google is often a good method of finding helpful material.

What is Unix?



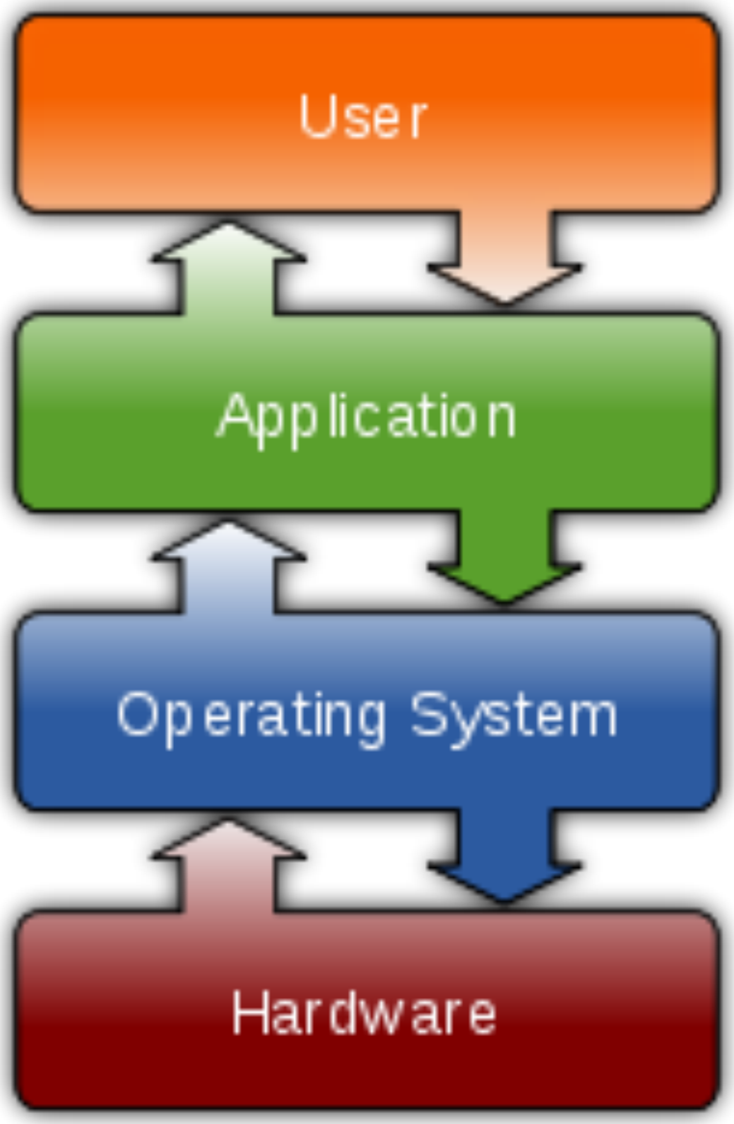
What is Unix?

- Unix is a ***class*** of operating systems (think Windows, OSX, Ubuntu)



What is Unix?

- Unix is a ***class*** of operating systems (think Windows, OSX, Ubuntu)
- an Operating system, it site between your applications (perl, Word, etc.) and your hardware



What is Unix?

What is Unix?

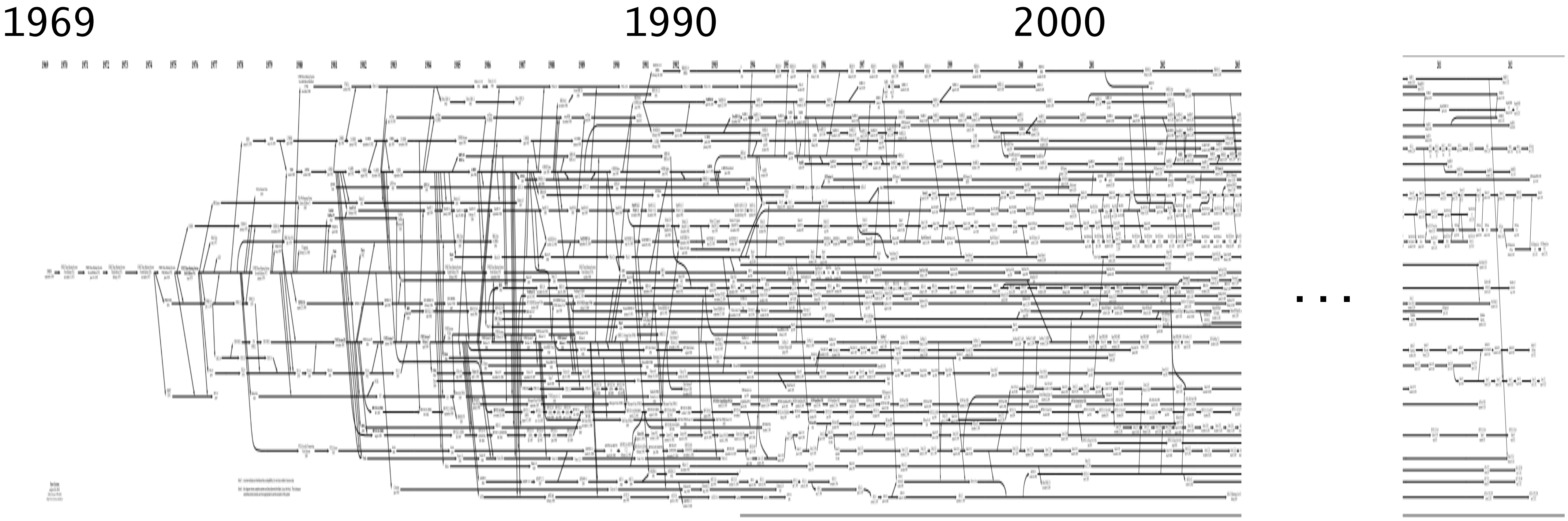
- In fact Ubuntu is a variant (or distribution) of Linux

What is Unix?

- In fact Ubuntu is a variant (or distribution) of Linux
- Linux as well as OSX are variants of Unix!!

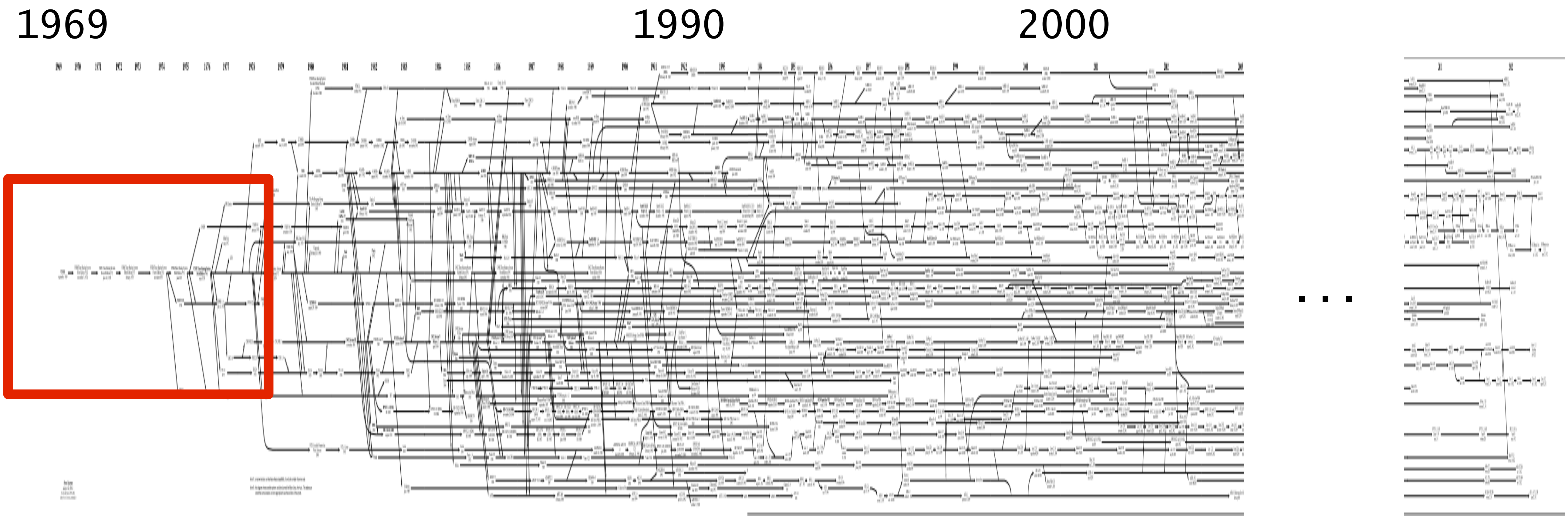
What is Unix?

- In fact Ubuntu is a variant (or distribution) of Linux
- Linux as well as OSX are variants of Unix!!

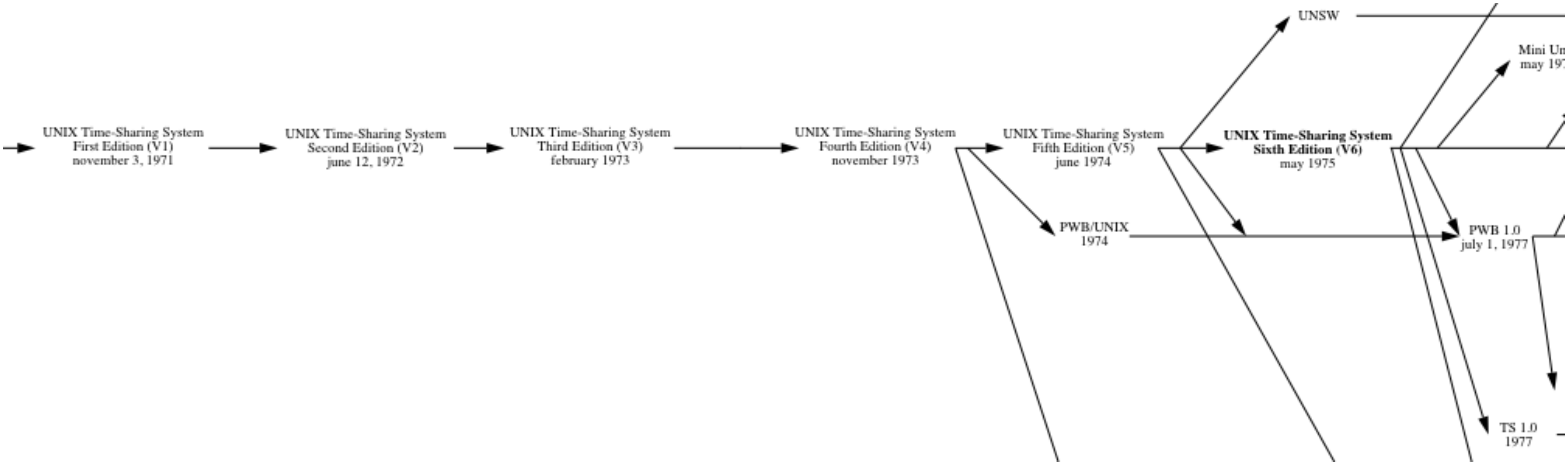


What is Unix?

- In fact Ubuntu is a variant (or distribution) of Linux
- Linux as well as OSX are variants of Unix!!



What is Unix?



UNIX Time-Sharing System
First Edition (V1)
november 3, 1971

UNIX Time-Sharing System
Second Edition (V2)
june 12, 1972

UNIX Time-Sharing System
Third Edition (V3)
february 1973

UNIX Time-Sharing System
Fourth Edition (V4)
november 1973

UNIX Time-Sharing System
Fifth Edition (V5)
june 1974

PWB/UNIX
1974

UNSW

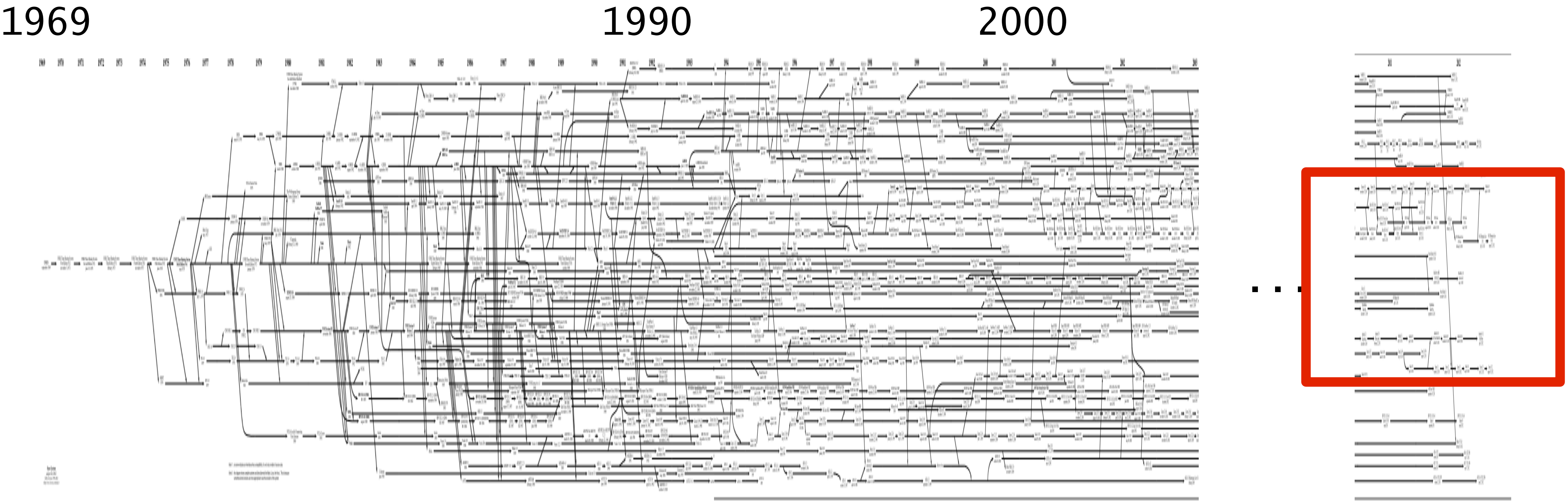
UNIX Time-Sharing System
Sixth Edition (V6)
may 1975

Mini Un
may 19

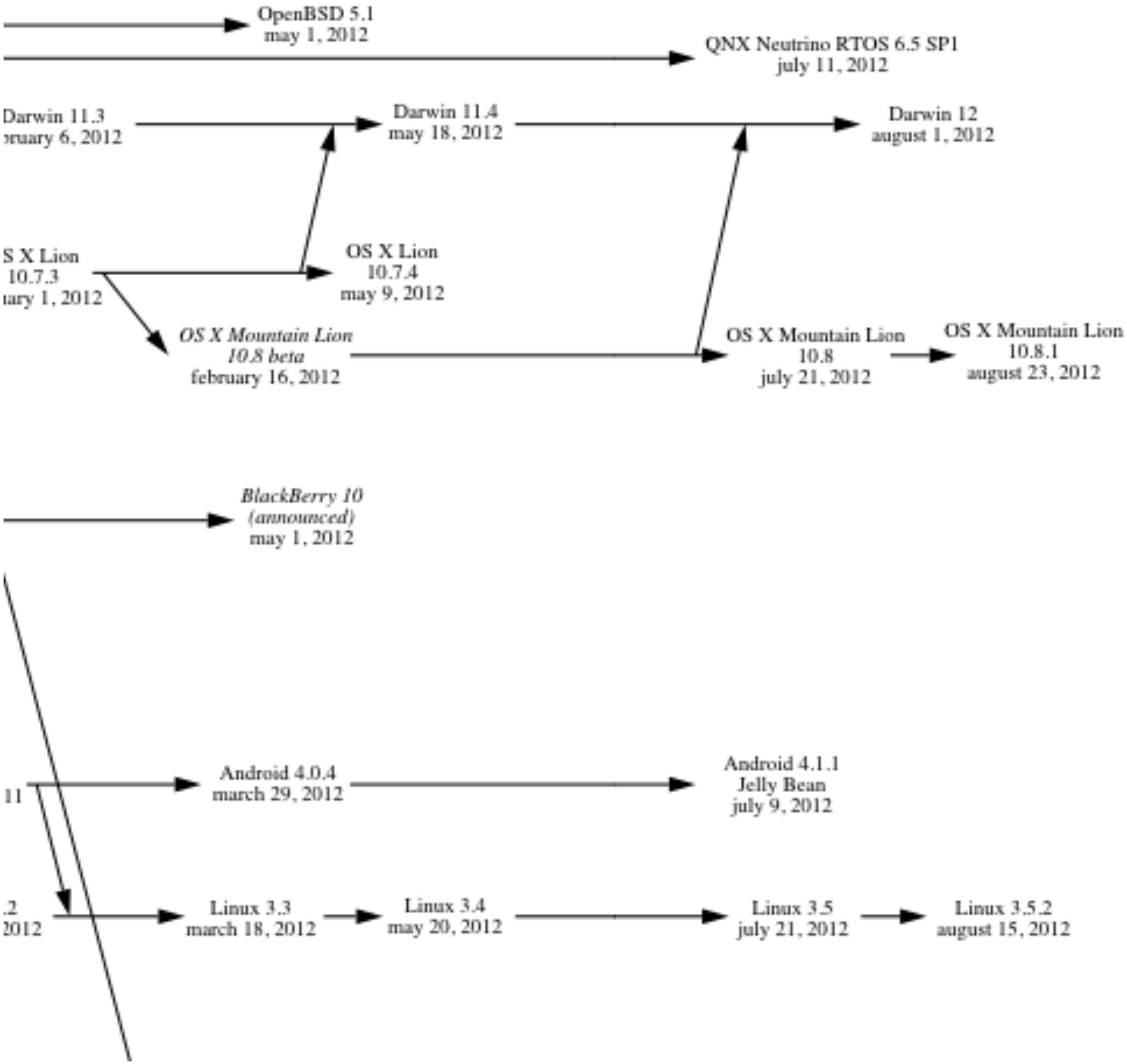
PWB 1.0
july 1, 1977

TS 1.0
1977

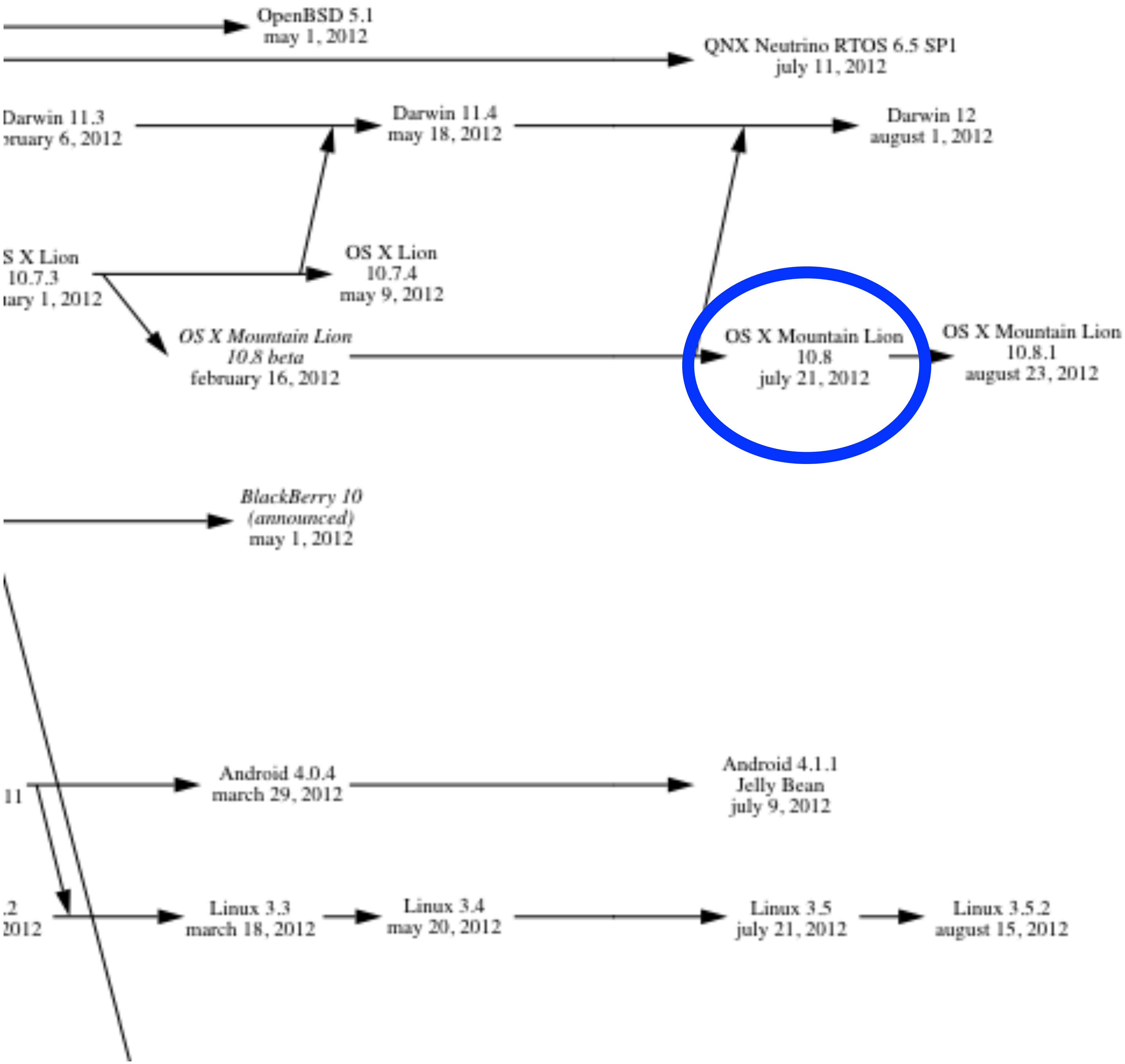
What is Unix?



What is Unix?


OpenBSD 5.1
may 1, 2012
QNX Neutrino RTOS 6.5 SP1
july 11, 2012
Darwin 11.3
›ruary 6, 2012
Darwin 11.4
may 18, 2012
Darwin 12
august 1, 2012
S X Lion
10.7.3
iary 1, 2012
OS X Lion
10.7.4
may 9, 2012
OS X Mountain Lion
10.8 beta
february 16, 2012
OS X Mountain Lion
10.8
july 21, 2012
OS X Mountain Lion
10.8.1
august 23, 2012
BlackBerry 10
(announced)
may 1, 2012
11
Android 4.0.4
march 29, 2012
Android 4.1.1
Jelly Bean
july 9, 2012
.2
2012
Linux 3.3
march 18, 2012
Linux 3.4
may 20, 2012
Linux 3.5
july 21, 2012
Linux 3.5.2
august 15, 2012

What is Unix?



OpenBSD 5.1
may 1, 2012

QNX Neutrino RTOS 6.5 SP1
july 11, 2012

Darwin 11.3
ruary 6, 2012

Darwin 11.4
may 18, 2012

Darwin 12
august 1, 2012

S X Lion
10.7.3
ary 1, 2012

OS X Lion
10.7.4
may 9, 2012

OS X Mountain Lion
10.8 beta
february 16, 2012

OS X Mountain Lion
10.8
july 21, 2012

OS X Mountain Lion
10.8.1
august 23, 2012

BlackBerry 10
(announced)
may 1, 2012

11

Android 4.0.4
march 29, 2012

Android 4.1.1
Jelly Bean
july 9, 2012

.2
2012

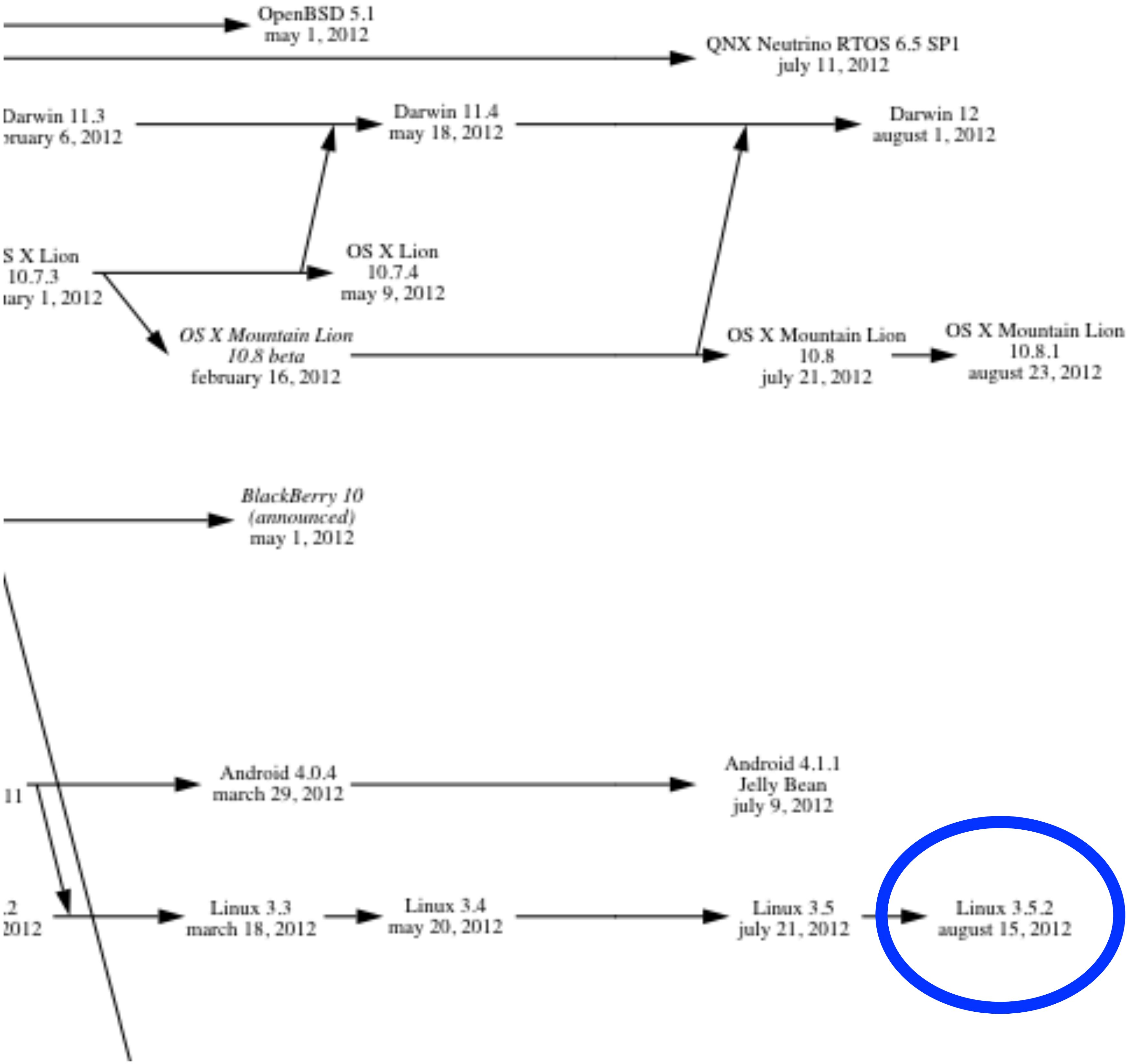
Linux 3.3
march 18, 2012

Linux 3.4
may 20, 2012

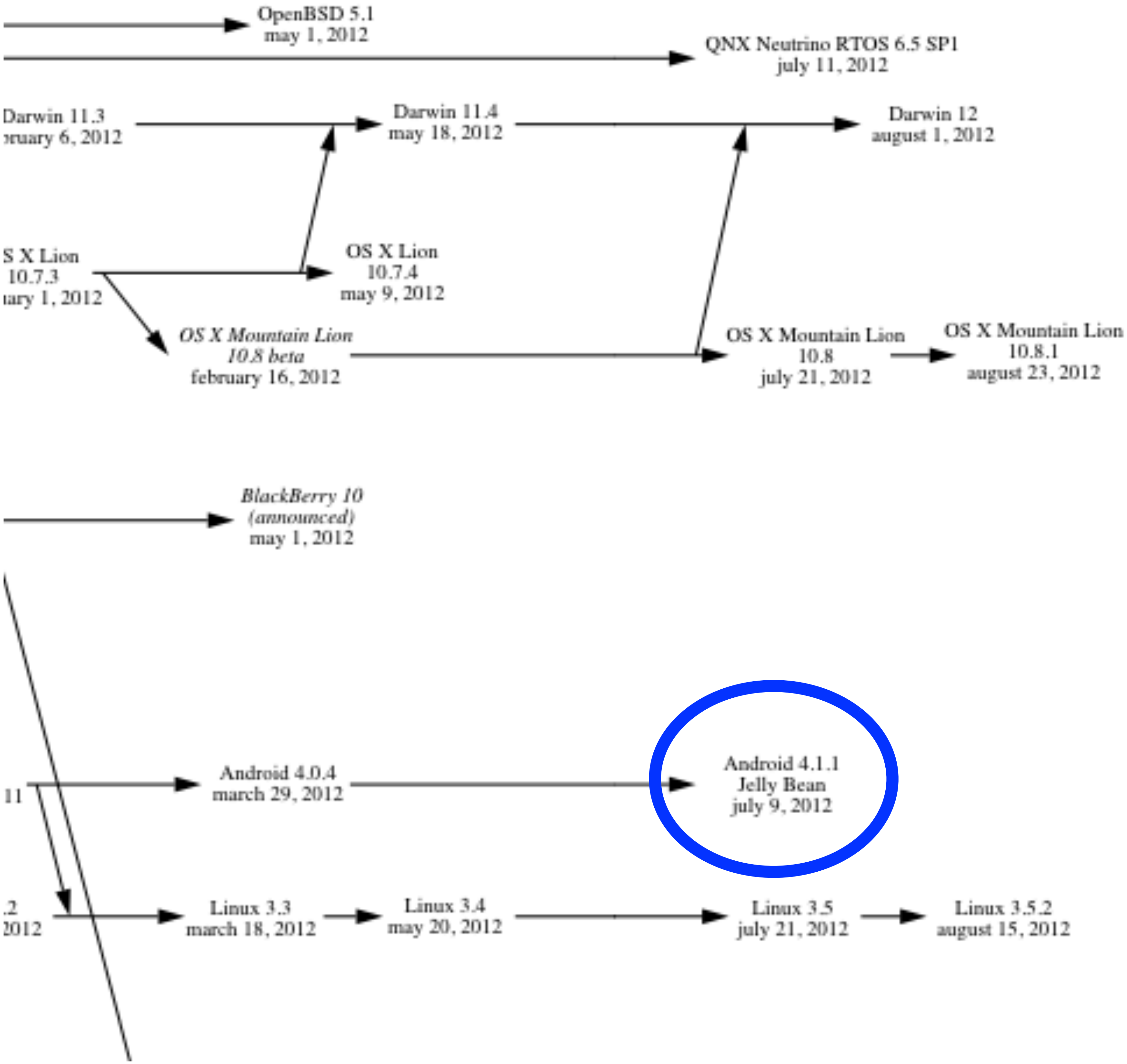
Linux 3.5
july 21, 2012

Linux 3.5.2
august 15, 2012

What is Unix?


OpenBSD 5.1
may 1, 2012
QNX Neutrino RTOS 6.5 SP1
july 11, 2012
Darwin 11.3
ruary 6, 2012
Darwin 11.4
may 18, 2012
Darwin 12
august 1, 2012
S X Lion
10.7.3
ary 1, 2012
OS X Lion
10.7.4
may 9, 2012
OS X Mountain Lion
10.8 beta
february 16, 2012
OS X Mountain Lion
10.8
july 21, 2012
OS X Mountain Lion
10.8.1
august 23, 2012
BlackBerry 10
(announced)
may 1, 2012
11
Android 4.0.4
march 29, 2012
Android 4.1.1
Jelly Bean
july 9, 2012
.2
2012
Linux 3.3
march 18, 2012
Linux 3.4
may 20, 2012
Linux 3.5
july 21, 2012
Linux 3.5.2
august 15, 2012

What is Unix?



OpenBSD 5.1
may 1, 2012

QNX Neutrino RTOS 6.5 SP1
july 11, 2012

Darwin 11.3
ruary 6, 2012

Darwin 11.4
may 18, 2012

Darwin 12
august 1, 2012

S X Lion
10.7.3
uary 1, 2012

OS X Lion
10.7.4
may 9, 2012

OS X Mountain Lion
10.8 beta
february 16, 2012

OS X Mountain Lion
10.8
july 21, 2012

OS X Mountain Lion
10.8.1
august 23, 2012

BlackBerry 10
(announced)
may 1, 2012

11

Android 4.0.4
march 29, 2012

Android 4.1.1
Jelly Bean
july 9, 2012

.2
2012

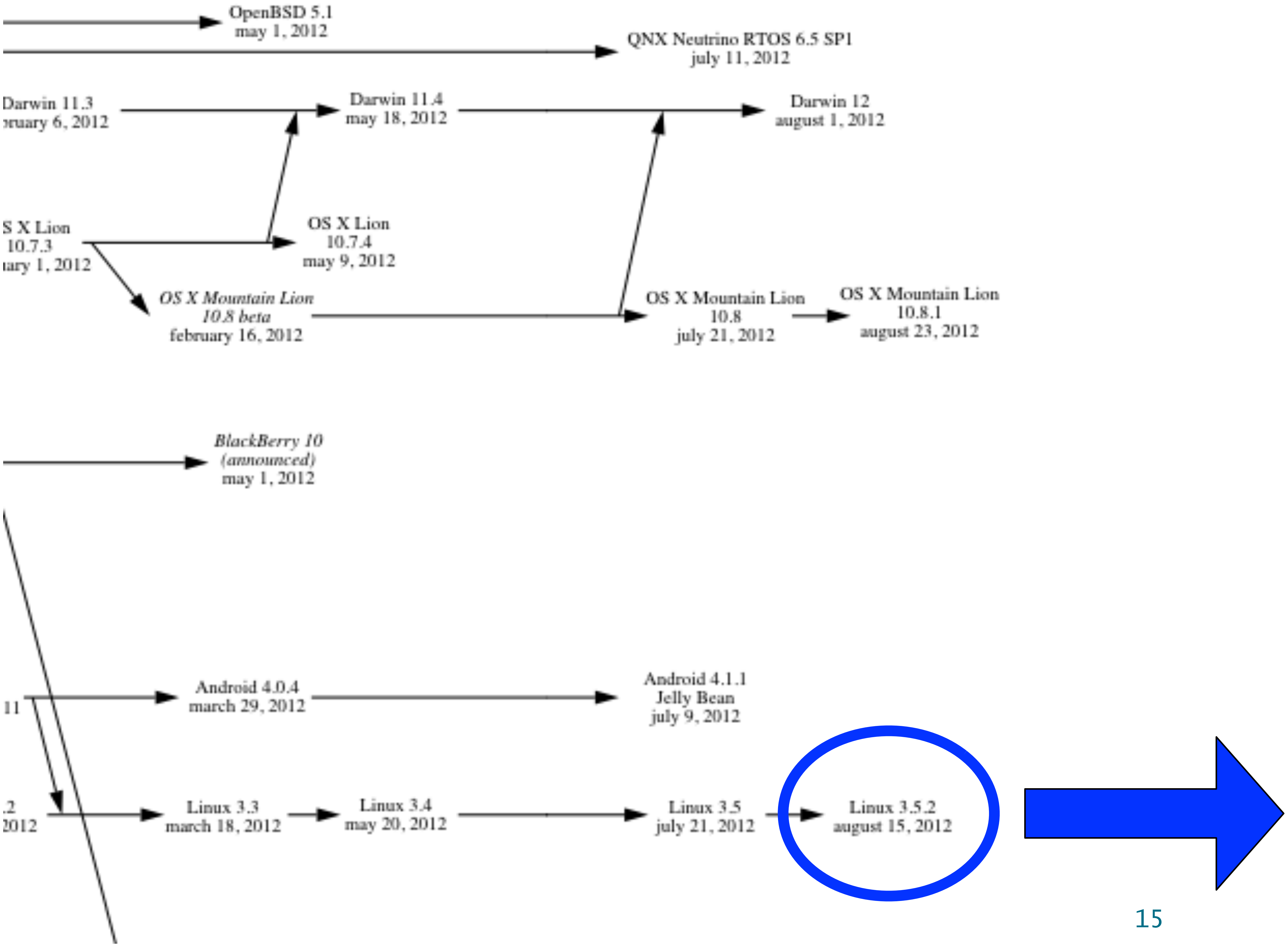
Linux 3.3
march 18, 2012

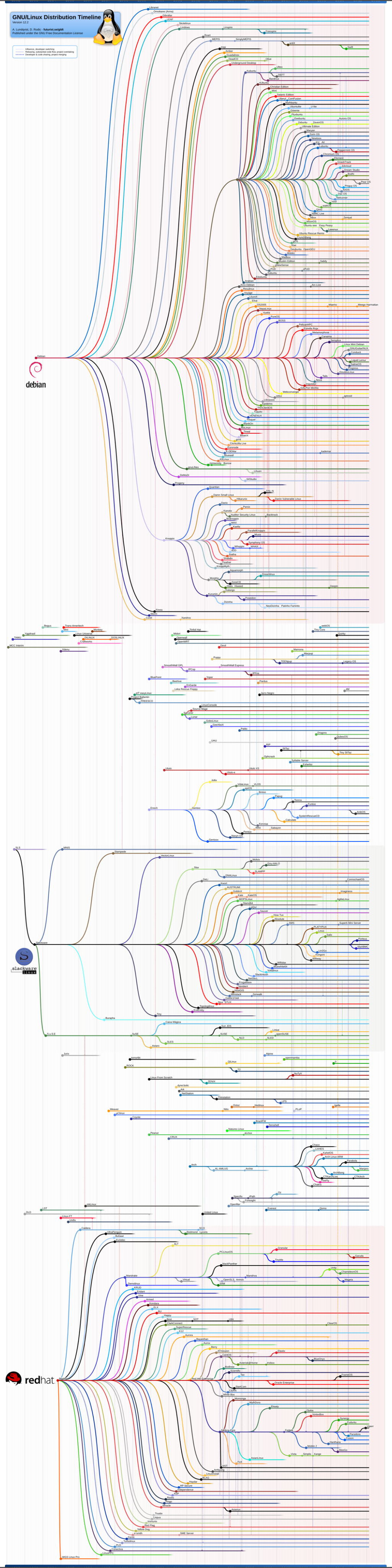
Linux 3.4
may 20, 2012

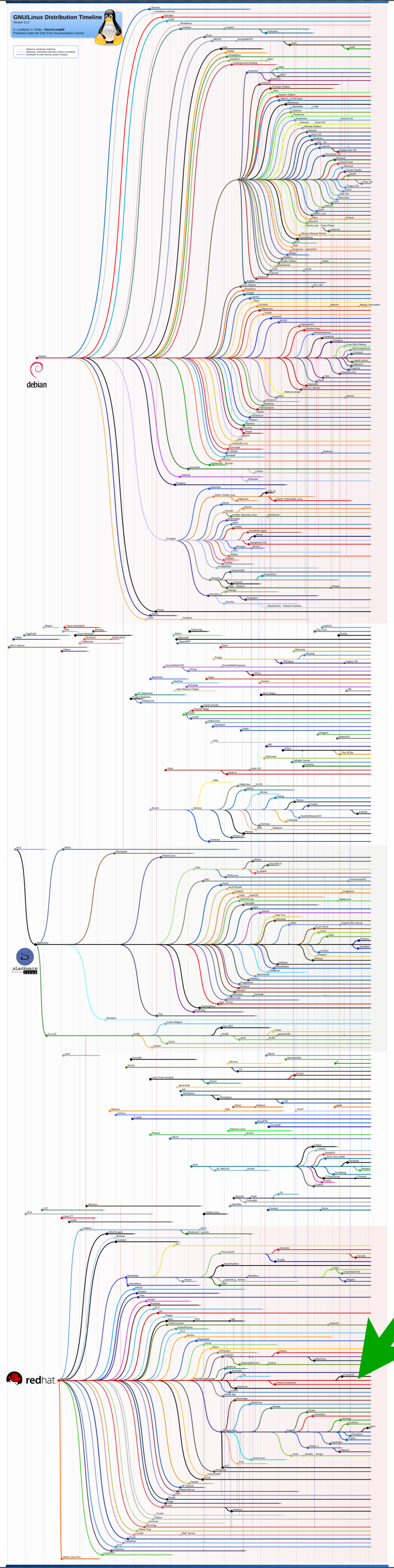
Linux 3.5
july 21, 2012

Linux 3.5.2
august 15, 2012

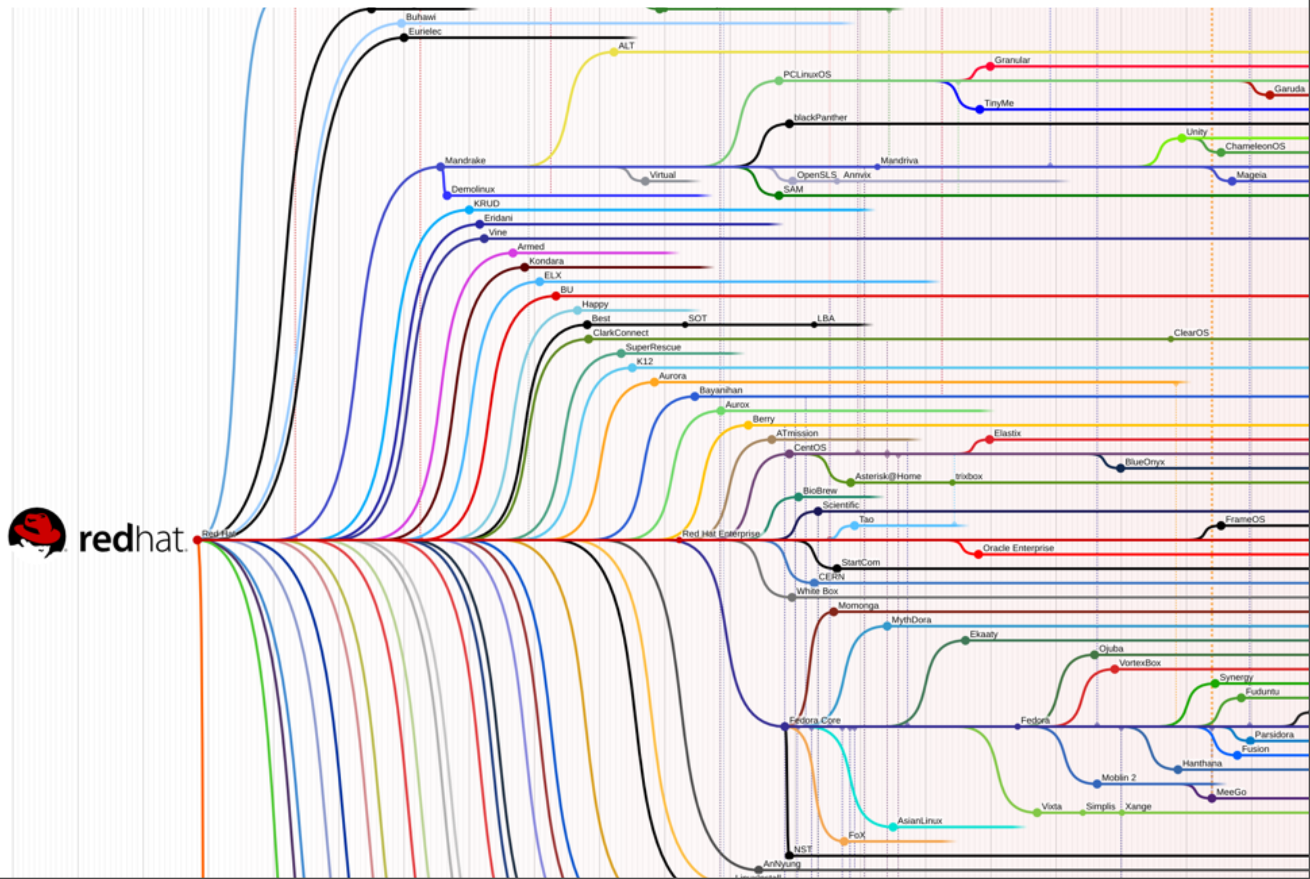
What is Unix?

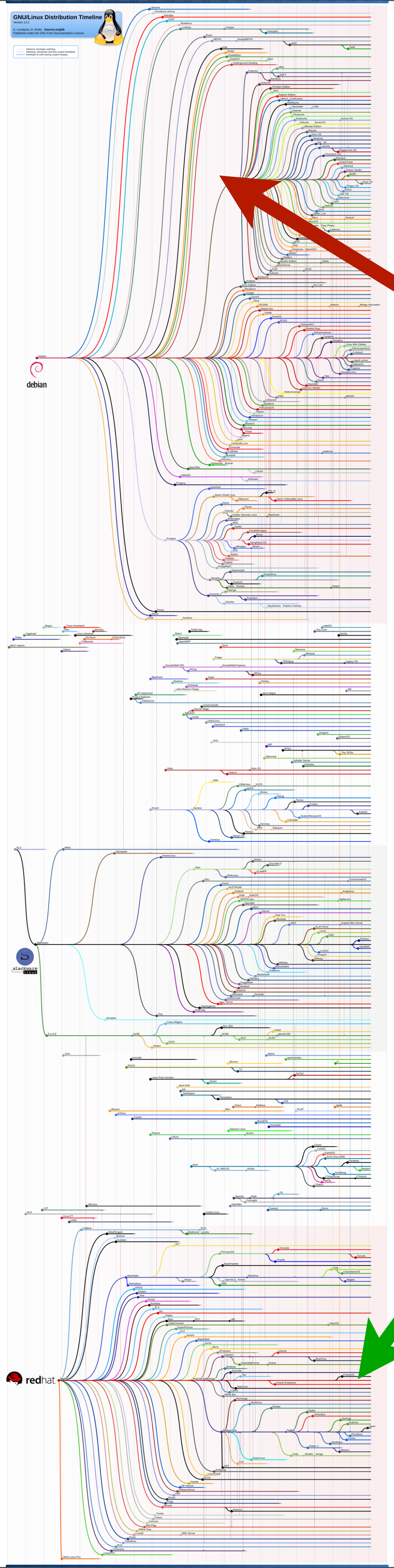

OpenBSD 5.1
may 1, 2012
QNX Neutrino RTOS 6.5 SP1
july 11, 2012
Darwin 11.3
ruary 6, 2012
Darwin 11.4
may 18, 2012
Darwin 12
august 1, 2012
S X Lion
10.7.3
ary 1, 2012
OS X Lion
10.7.4
may 9, 2012
OS X Mountain Lion
10.8 beta
february 16, 2012
OS X Mountain Lion
10.8
july 21, 2012
OS X Mountain Lion
10.8.1
august 23, 2012
BlackBerry 10
(announced)
may 1, 2012
11
Android 4.0.4
march 29, 2012
Android 4.1.1
Jelly Bean
july 9, 2012
.2
2012
Linux 3.3
march 18, 2012
Linux 3.4
may 20, 2012
Linux 3.5
july 21, 2012
Linux 3.5.2
august 15, 2012


GNU/Linux Distribution Timeline
Version 12.2
A. Lundqvist, D. Rodic - futurist.se/gldt
Published under the GNU Free Documentation License
Influence, developer switching
Rebasing, substantial code flow, project overtaking
Developer & code sharing, project merging
Libranet
debian
slackware
redhat


GNU/Linux Distribution Timeline
debian
redhat

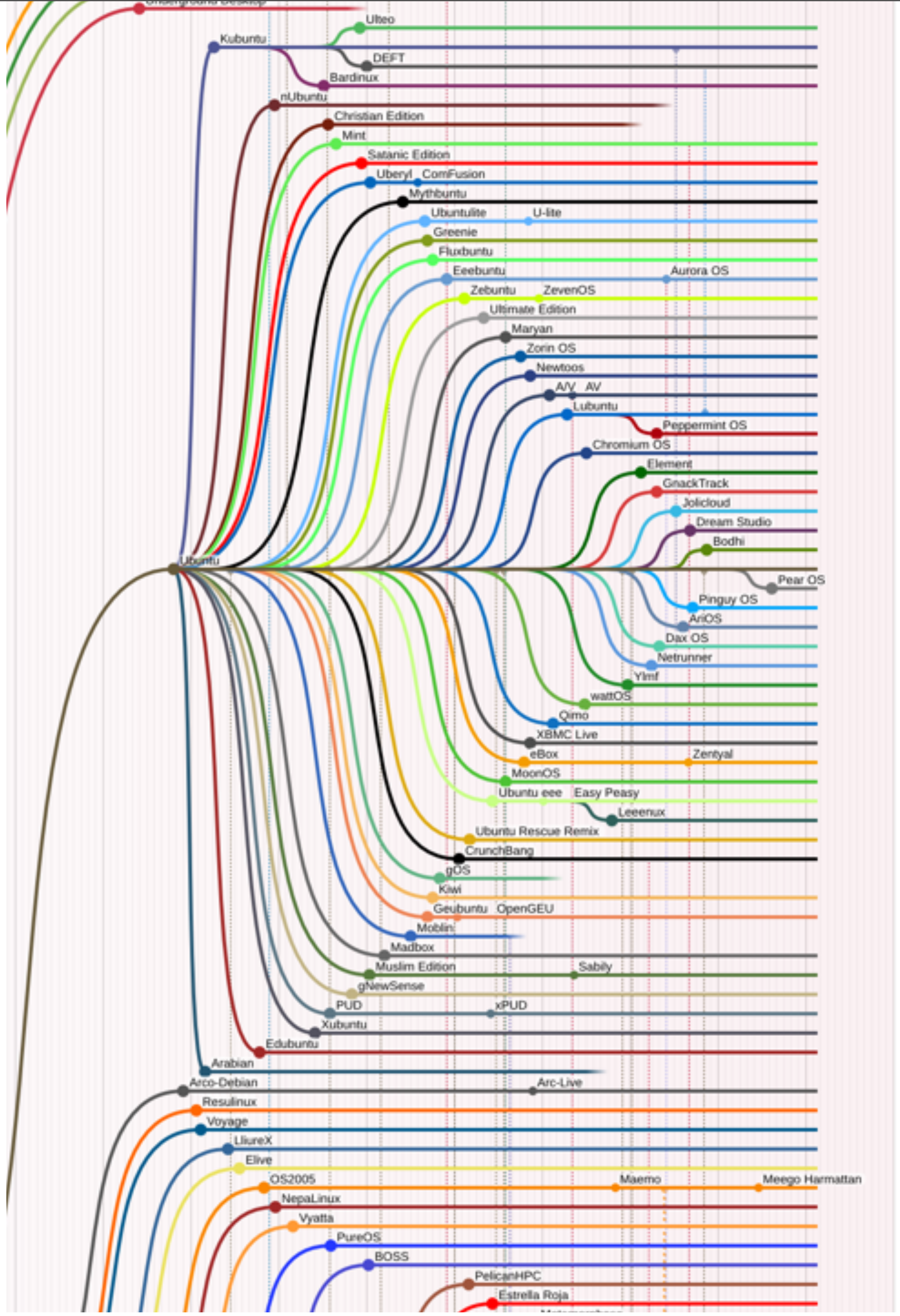
UA HPC


Buhawi
Eurielec
ALT
PCLinuxOS
Granular
Garuda
TinyMe
blackPanther
Unity
ChameleonOS
Mandrake
Mandriva
Virtual
OpenSLS
Annvix
Mageia
Demolinux
SAM
KRUD
Eridani
Vine
Armed
Kondara
ELX
BU
Happy
Best
SOT
LBA
ClarkConnect
ClearOS
SuperRescue
K12
Aurora
Bayanihan
Aurox
Berry
ATmission
Elastix
CentOS
BlueOnyx
Asterisk@Home
trixbox
BioBrew
Scientific
Tao
FrameOS
redhat
Red Hat
Red Hat Enterprise
Oracle Enterprise
StartCom
CERN
White Box
Momonga
MythDora
Ekaaty
Ojuba
VortexBox
Synergy
Fuduntu
Fedora Core
Fedora
Parsidora
Fusion
Hanthana
Moblin 2
MeeGo
Vixta
Simplis
Xange
AsianLinux
FoX
NST
AnNyung



Ubuntu

UA HPC



What is Unix?

What is Unix?

What is Unix?

- If you want a Linux-like environment on a Windows system, you can download and install cygwin: http://cygwin.com/

What is Unix?

- If you want a Linux-like environment on a Windows system, you can download and install cygwin: http://cygwin.com/

What is Unix?

- If you want a Linux-like environment on a Windows system, you can download and install cygwin: http://cygwin.com/
- On top of the Unix operating system kernel is a Unix shell (sh, csh, tcsh, ksh, bash, etc.) The shell interprets the commands you type and interacts with the appropriate operating system components

UNIX File System

UNIX File System

- Files and directories are organized hierarchically:

UNIX File System

- Files and directories are organized hierarchically:
- The Unix file system is an inverted tree of files and directories (the root is at the top). Files are like the leaves on the tree.

UNIX File System

- Files and directories are organized hierarchically:
- The Unix file system is an inverted tree of files and directories (the root is at the top). Files are like the leaves on the tree.
- The 'root' is referenced by a forward slash /

UNIX File System

- Files and directories are organized hierarchically:
- The Unix file system is an inverted tree of files and directories (the root is at the top). Files are like the leaves on the tree.
- The 'root' is referenced by a forward slash /
- Each file has an absolute path from the root that begins with the forward slash. Files can also be referenced using relative paths, which depend upon where in the file system you are currently working.

UNIX File System

- Files and directories are organized hierarchically:
- The Unix file system is an inverted tree of files and directories (the root is at the top). Files are like the leaves on the tree.
- The 'root' is referenced by a forward slash /
- Each file has an absolute path from the root that begins with the forward slash. Files can also be referenced using relative paths, which depend upon where in the file system you are currently working.
- Subdirectories in a path are separated by forward slash

UNIX File System

- Files and directories are organized hierarchically:
- The Unix file system is an inverted tree of files and directories (the root is at the top). Files are like the leaves on the tree.
- The 'root' is referenced by a forward slash /
- Each file has an absolute path from the root that begins with the forward slash. Files can also be referenced using relative paths, which depend upon where in the file system you are currently working.
- Subdirectories in a path are separated by forward slash
- Examples:
  - /
  - /home
  - /home/plantsci/sam/data/samples
  - /usr
  - /usr/local/bin
  - data/links
  - data/samples

UNIX File System

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)

- At any given time, you are working inside a directory

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)
- At any given time, you are working inside a directory
- You have a home directory – this is your starting point. Your home directory has an absolute path, such as /home/users/joe, but this can be referred to as ~ (tilde)

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)
- At any given time, you are working inside a directory
- You have a home directory – this is your starting point. Your home directory has an absolute path, such as /home/users/joe, but this can be referred to as ~ (tilde)
- Your current directory is referred to as . (dot)

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)
- At any given time, you are working inside a directory
- You have a home directory – this is your starting point. Your home directory has an absolute path, such as /home/users/joe, but this can be referred to as ~ (tilde)
- Your current directory is referred to as . (dot)
- Your parent directory (one level above) is .. (dot dot)

UNIX File System

- Files are organized into directories (aka folders on non-Unix systems)
- At any given time, you are working inside a directory
- You have a home directory – this is your starting point. Your home directory has an absolute path, such as /home/users/joe, but this can be referred to as ~ (tilde)
- Your current directory is referred to as . (dot)
- Your parent directory (one level above) is .. (dot dot)
- The pwd (present working directory) command will tell you which directory you are in. On some computer systems the command prompt reflects the working directory.

UNIX File Names

UNIX File Names

- UNIX is case sensitive!

UNIX File Names

- UNIX is case sensitive!
- Filenames should not contain spaces or punctuation other than a period or dot . (especially not *&!%$|^/\~)

UNIX File Names

- UNIX is case sensitive!
- Filenames should not contain spaces or punctuation other than a period or dot . (especially not *&!%$|^/\~)
- Filename “extensions” are optional, and usually suggest the file type.

UNIX File Names

- UNIX is case sensitive!
- Filenames should not contain spaces or punctuation other than a period or dot . (especially not *&!%$|^/\~)
- Filename “extensions” are optional, and usually suggest the file type.
- Underscores can be used instead of spaces in filenames

UNIX File Names

- UNIX is case sensitive!
- Filenames should not contain spaces or punctuation other than a period or dot . (especially not *&!%$|^/\~)
- Filename “extensions” are optional, and usually suggest the file type.
- Underscores can be used instead of spaces in filenames
- Examples:
  - `Sample_Data`
  - `sampleData.22.02.07`
  - `my_demo.txt`
  - `A01_JR165_T7.ab1`
  - `index.html`

Introduction to UNIX commands

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!
- Be sure to type a space between the command and each of the options and arguments!!

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!
- Be sure to type a space between the command and each of the options and arguments!!
- Typically options are preceded by a dash –

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!
- Be sure to type a space between the command and each of the options and arguments!!
- Typically options are preceded by a dash –
- When typing options on the command line, do NOT type the square brackets

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!
- Be sure to type a space between the command and each of the options and arguments!!
- Typically options are preceded by a dash –
- When typing options on the command line, do NOT type the square brackets
- The command will execute when you press the Enter key

Introduction to UNIX commands

- Unix commands can be terse, but are powerful!
- Command format: (VERY IMPORTANT!!)

```
command    [-options]    arguments
```

- Remember that UNIX is Case Sensitive! Remember the Command format – it never changes!
- Be sure to type a space between the command and each of the options and arguments!!
- Typically options are preceded by a dash –
- When typing options on the command line, do NOT type the square brackets
- The command will execute when you press the Enter key
- If you do not see an error message, the command executed without a problem (Unix is not chatty!)

Logging in to the HPC system

Logging in to the HPC system

- Open the Terminal application
  - On Mac/Linux, Terminal is in Applications-->Utilities
  - On Windows, use PuTTY (find it on google)

- Type:

```
ssh <myUANetID>@login.hpc.arizona.edu
```

- You should see a command prompt such as:

```
[service0][~]>
```

OR

```
/uNN/netid>
```

Understanding Command Examples

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.
- Example:

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir <my_dir>`

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir  <my_dir>`
    - The mkdir command will create a directory with the name given as an argument. The name *`my_dir`* is not a good name because it is not descriptive! Give directories meaningful names, e.g:

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir  <my_dir>`
    - The mkdir command will create a directory with the name given as an argument. The name *`my_dir`* is not a good name because it is not descriptive! Give directories meaningful names, e.g:

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir  <my_dir>`
    - The mkdir command will create a directory with the name given as an argument. The name *`my_dir`* is not a good name because it is not descriptive! Give directories meaningful names, e.g:

```
mkdir  SNP_project
```

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir  <my_dir>`
    - The mkdir command will create a directory with the name given as an argument. The name *`my_dir`* is not a good name because it is not descriptive! Give directories meaningful names, e.g:

```
mkdir  SNP_project
```

What about this?

Understanding Command Examples

- In these slides, when you see command arguments in italics, do not type the <*italicized bracketed*> words literally. Replace them with the appropriate arguments.

- Example:
  - `mkdir <my_dir>`
    - The mkdir command will create a directory with the name given as an argument. The name *`my_dir`* is not a good name because it is not descriptive! Give directories meaningful names, e.g:

```
mkdir  SNP_project
```

What about this?

```
mkdir  Human SNP project
```

Understanding Command Examples

Understanding Command Examples

- A special example of user input is `<fileList>`

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

Understanding Command Examples

- A special example of user input is `<fileList>`
    - this is one or more files from the filesystem
    - typically, multiple files would be specified with the *

- Some examples are

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

- Some examples are
  - `ls file1 file2`

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

- Some examples are
  - `ls file1 file2`
  - `cat chrom*`

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

- Some examples are
  - `ls file1 file2`
  - `cat chrom*`
  - `vim *.fa`

Understanding Command Examples

- A special example of user input is `<fileList>`
  - this is one or more files from the filesystem
  - typically, multiple files would be specified with the *

- Some examples are
  - `ls file1 file2`
  - `cat chrom*`
  - `vim *.fa`
  - `tail /tmp/ecol553/*.fa`

Command Options and Arguments

Command Options and Arguments

- Some commands require arguments, others do not.

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.
- If you type a command with an incorrect number of arguments, Unix will display a 'usage' message to help you out.

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.
- If you type a command with an incorrect number of arguments, Unix will display a 'usage' message to help you out.
- Command options are not required, but if used are typically typed before the arguments

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.
- If you type a command with an incorrect number of arguments, Unix will display a 'usage' message to help you out.
- Command options are not required, but if used are typically typed before the arguments
  - Examples:

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.
- If you type a command with an incorrect number of arguments, Unix will display a 'usage' message to help you out.
- Command options are not required, but if used are typically typed before the arguments
  - Examples:
    - cp fileA fileB

Command Options and Arguments

- Some commands require arguments, others do not.
- Some commands can handle any number of arguments.
- If you type a command with an incorrect number of arguments, Unix will display a 'usage' message to help you out.
- Command options are not required, but if used are typically typed before the arguments
  - Examples:
    - cp fileA fileB
    - cp –r /genome/student/data testdata

Examining the `ls` command

Examining the `ls` command

- The `ls` command lists the contents of a directory

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
    - `-l` long listing
      includes owner, permissions, last modified date, etc.

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order
  - `-F` show File type
    marks directories with /, executable files with *

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
    - `-l` long listing
      includes owner, permissions, last modified date, etc.
    - `-r` reverses the sort order
    - `-F` show File type
      marks directories with /, executable files with *
    - `-t` sorts by time last modified

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:

`-l`	long listing includes owner, permissions, last modified date, etc.
`-r`	reverses the sort order
`-F`	show File type marks directories with /, executable files with *
`-t`	sorts by time last modified

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    - includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order
  - `-F` show File type
    - marks directories with /, executable files with *
  - `-t` sorts by time last modified

  - Examples:

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order
  - `-F` show File type
    marks directories with /, executable files with *
  - `-t` sorts by time last modified

  - Examples:
    - `ls   -l`

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order
  - `-F` show File type
    marks directories with /, executable files with *
  - `-t` sorts by time last modified

  - Examples:
    - `ls  -l`
    - `ls  -ltr`

Examining the `ls` command

- The `ls` command lists the contents of a directory
- With no arguments, the current directory is listed
- There are many options for `ls`, and they can be combined:
  - `-l` long listing
    - includes owner, permissions, last modified date, etc.
  - `-r` reverses the sort order
  - `-F` show File type
    - marks directories with /, executable files with *
  - `-t` sorts by time last modified

  - Examples:
    - `ls   -l`
    - `ls   -ltr`
    - `ls   -l   chromat_dir`

Getting help for UNIX commands

The man command (short for manual) can be used to see documentation for any UNIX command (even man itself!) To do a keyword search of the documentation, use man –k keyword

Example: `man man`

```
User Commands                                         man(1)

NAME
     man - find and display reference manual pages

SYNOPSIS
     man [ - ]  [ -adFlrt ]  [ -M path ]  [ -T macro-package ]  [
     -s section ]  name ...

     man [ -M path ]  -k keyword ...

DESCRIPTION
     The man command  displays  information  from  the  reference
     manuals.  It  displays complete manual pages that you select
     by name, or one-line summaries selected  either  by  keyword
     (-k),  or  by  the  name  of  an associated file (-f). If no
     manual page is located, man prints an error message.   …
```

The U of A Supercomputers

The U of A Supercomputers

- The campus Supercomputers allow you to run jobs on many CPUs simultaneously.

The U of A Supercomputers

- The campus Supercomputers allow you to run jobs on many CPUs simultaneously.

- For example, if you need to run BLAST on 1600 sequences, these could be divided up into 16 batches of 100 sequences each and run on 16 processors. The runtime would be similar to blasting 100 sequences with a single processor.

The U of A Supercomputers

- The campus Supercomputers allow you to run jobs on many CPUs simultaneously.

- For example, if you need to run BLAST on 1600 sequences, these could be divided up into 16 batches of 100 sequences each and run on 16 processors. The runtime would be similar to blasting 100 sequences with a single processor.

- There are three different supercomputers. The shared memory system is '`smp`', and the cluster system is '`cluster`', and the high throughput cluster is ‘`htc`’. All of these can be accessed from “`login`” (`login.hpc.arizona.edu`)

The U of A Supercomputers

- The campus Supercomputers allow you to run jobs on many CPUs simultaneously.

- For example, if you need to run BLAST on 1600 sequences, these could be divided up into 16 batches of 100 sequences each and run on 16 processors. The runtime would be similar to blasting 100 sequences with a single processor.

- There are three different supercomputers. The shared memory system is '`smp`', and the cluster system is '`cluster`', and the high throughput cluster is ‘`htc`’. All of these can be accessed from “`login`” (`login.hpc.arizona.edu`)

- FAQs and User Guides are available here:
  http://rc.arizona.edu/hpc-htc/using-systems

Copying files

Copying files

- The `cp` command copies files or directories from one place to another. The files you want to copy are called "sources" and the place you want to copy them to is called the "destination". To copy an entire directory, use the `–r` option.
- Examples:
  - `cp   fileA   filA.backup`
  - `cp   fileA   fileB   recycle_bin/`
  - `cp –r   recycle_bin/   backup_dir/`

- Remember that `cp` ALWAYS requires *at least* two arguments: the source files/directories and the destination file/directory (often this will be .)

- Use the `cp` command to copy a directory of files to work with:
  - `cp –r /genome/student/data playground`

Exercise: practice with `man`, `pwd`, and `ls`

Exercise: practice with `man`, `pwd`, and `ls`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Type `ls –la`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Type `ls –la`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Type `ls –la`

Type `ls –l playground`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Type `ls –la`

Type `ls –l playground`

Exercise: practice with `man`, `pwd`, and `ls`

Type `pwd`

Type `LS`

Type `ls`

Type `man`

Type `man ls`

Type `ls –la`

Type `ls –l playground`

When finished, type `logout`

UNIX Basic Command Recap

ssh [<login>@]*<system_address>*	Log in to a remote computer
man *<command>*	Output a manual page for the specified command
pwd	Display the present working directory
mkdir *<dirname>* …	Make new directory/directories named dirname …
ls *[<dirname> …]* ls -l	List directory contents (of dirname(s)) Show a long listing of directory contents
cp *<srcfile>* … *<dest>* cp –r *<srcdir>* … *<dest>*	Copy source file(s) to destination Recursively copy source dir(s) to destination
logout	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]`	List directory contents (of dirname(s))
`ls -l`	Show a long listing of directory contents
`cp <srcfile> … <dest>`	Copy source file(s) to destination
`cp –r <srcdir> … <dest>`	Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]`*<system_address>*	Log in to a remote computer
`man` *<command>*	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir` *<dirname> …*	Make new directory/directories named dirname …
`ls` *[<dirname> …]* `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp` *<srcfile> … <dest>* `cp –r` *<srcdir> … <dest>*	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp —r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh  [<login>@]<system_address>`	Log in to a remote computer
`man  <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir  <dirname> …`	Make new directory/directories named dirname …
`ls  [<dirname> …]` `ls  -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp  <srcfile> …  <dest>` `cp –r  <srcdir> …  <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname ...
`ls [<dirname> …]`	List directory contents (of dirname(s))
`ls -l`	Show a long listing of directory contents
`cp <srcfile> … <dest>`	Copy source file(s) to destination
`cp —r <srcdir> … <dest>`	Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname …
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

UNIX Basic Command Recap

- So far we have learned 7 basic commands:

`ssh [<login>@]<system_address>`	Log in to a remote computer
`man <command>`	Output a manual page for the specified command
`pwd`	Display the present working directory
`mkdir <dirname> …`	Make new directory/directories named dirname ...
`ls [<dirname> …]` `ls -l`	List directory contents (of dirname(s)) Show a long listing of directory contents
`cp <srcfile> … <dest>` `cp –r <srcdir> … <dest>`	Copy source file(s) to destination Recursively copy source dir(s) to destination
`logout`	Log out of the Unix system

- How many arguments does each of these commands take?

Using `wc`

Using `wc`

- wc = word count

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

A **word** is defined as a run of characters between whitespace (space, tab, newline)

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

A **line** is defined as the characters before a newline (i.e. the line count is the number of newlines)

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

A **byte** is 8 *bits*, or the amount of space necessary to store one character. So the byte count of a text file is the number of characters (including newlines, tabs, and spaces)

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

tab-delimited, or **tab-separated** format is a way of representing data where each record is on a new line, and entry related to that record is separated by a tab (‘\t’). An alternate is **comma-separated (csv).**

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

```
wc [<options>] <fileList>
```

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

```
wc [<options>] <fileList>
```

- a useful option is:

Using `wc`

- wc = word count
- returns the number of words, lines, and bytes in a file in tab-delimited format

```
wc [<options>] <fileList>
```

- a useful option is:
  - `-l` counts only the number lines

How to edit files (`VIM`)

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes
  - command mode (where you are when you start)

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes
  - command mode (where you are when you start)
  - insert mode

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes
  - command mode (where you are when you start)
  - insert mode
- to get from edit mode to command mode use *esc*

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes
  - command mode (where you are when you start)
  - insert mode
- to get from edit mode to command mode use *esc*
- to get to insert mode just type and *i*

How to edit files (`VIM`)

- the command to run `vim` is

```
vim <filename> [<fileList>]
```

- there are two modes
  - command mode (where you are when you start)
  - insert mode
- to get from edit mode to command mode use *esc*
- to get to insert mode just type and *i*
- to write the current file and exit type *:wq*

Useful VIM commands

Useful VIM commands

- to search a file

```
/<search>
```

Useful VIM commands

- to search a file

```
/<search>
```

- to find/replace

```
:%s/<search>/<replace>/[g]
```

Useful VIM commands

- to search a file

```
/<search>
```

- to find/replace

```
:%s/<search>/<replace>/[g]
```

These two commands can take *Regular Expressions* which we will talk about later

Useful VIM commands

- to search a file

```
/<search>
```

- to find/replace

```
:%s/<search>/<replace>/[g]
```

- to go to a line number

```
:<number>
```

Using `grep`

Using `grep`

- the command grep allows you to search a file without opening it

Using `grep`

- the command grep allows you to search a file without opening it

Using `grep`

- the command grep allows you to search a file without opening it

Using `grep`

- the command grep allows you to search a file without opening it

- some useful options are:

Using `grep`

- the command grep allows you to search a file without opening it

- some useful options are:
  - `-c` counts the number of occurrences of search

Using `grep`

- the command grep allows you to search a file without opening it

```
grep [<options>] <search> <fileList>
```

- some useful options are:
  - `-c` counts the number of occurrences of search
  - `-v` returns lines **without** search

Setting up environment

Setting up environment

- change bash to the default shell
  - `chsh -s /bin/bash`
  - type your password

Setting up environment

- change bash to the default shell
  - `chsh -s /bin/bash`
  - type your password
- edit your `~/.profile` file
  - add the following line:
    - `export PATH=~deblasio/bin/553:~/bin:$PATH`
    - `export PATH=/genome/ICEbin/:$PATH`

Setting up environment

- change bash to the default shell
  - `chsh -s /bin/bash`
  - type your password
- edit your `~/.profile` file
  - add the following line:
    - `export PATH=~deblasio/bin/553:~/bin:$PATH`
    - `export PATH=/genome/ICEbin/:$PATH`
- logout then log back into HPC

Setting up environment

- change bash to the default shell
  - `chsh -s /bin/bash`
  - type your password
- edit your `~/.profile` file
  - add the following line:
    - `export PATH=~deblasio/bin/553:~/bin:$PATH`
    - `export PATH=/genome/ICEbin/:$PATH`
- logout then log back into HPC
- try running `turnin --verify`

Setting up environment

- change bash to the default shell
  - `chsh -s /bin/bash`
  - type your password
- edit your `~/.profile` file
  - add the following line:
    - `export PATH=~deblasio/bin/553:~/bin:$PATH`
    - `export PATH=/genome/ICEbin/:$PATH`
- logout then log back into HPC
- try running `turnin --verify`
- You should see

```
deblasio@service0(login):~> turnin --verify
You are: deblasio
and you are good to go
```

Other helpful things for your `.profile`

Other helpful things for your `.profile`

- `export PS1="\u@\h(login):\w> "`

Other helpful things for your `.profile`

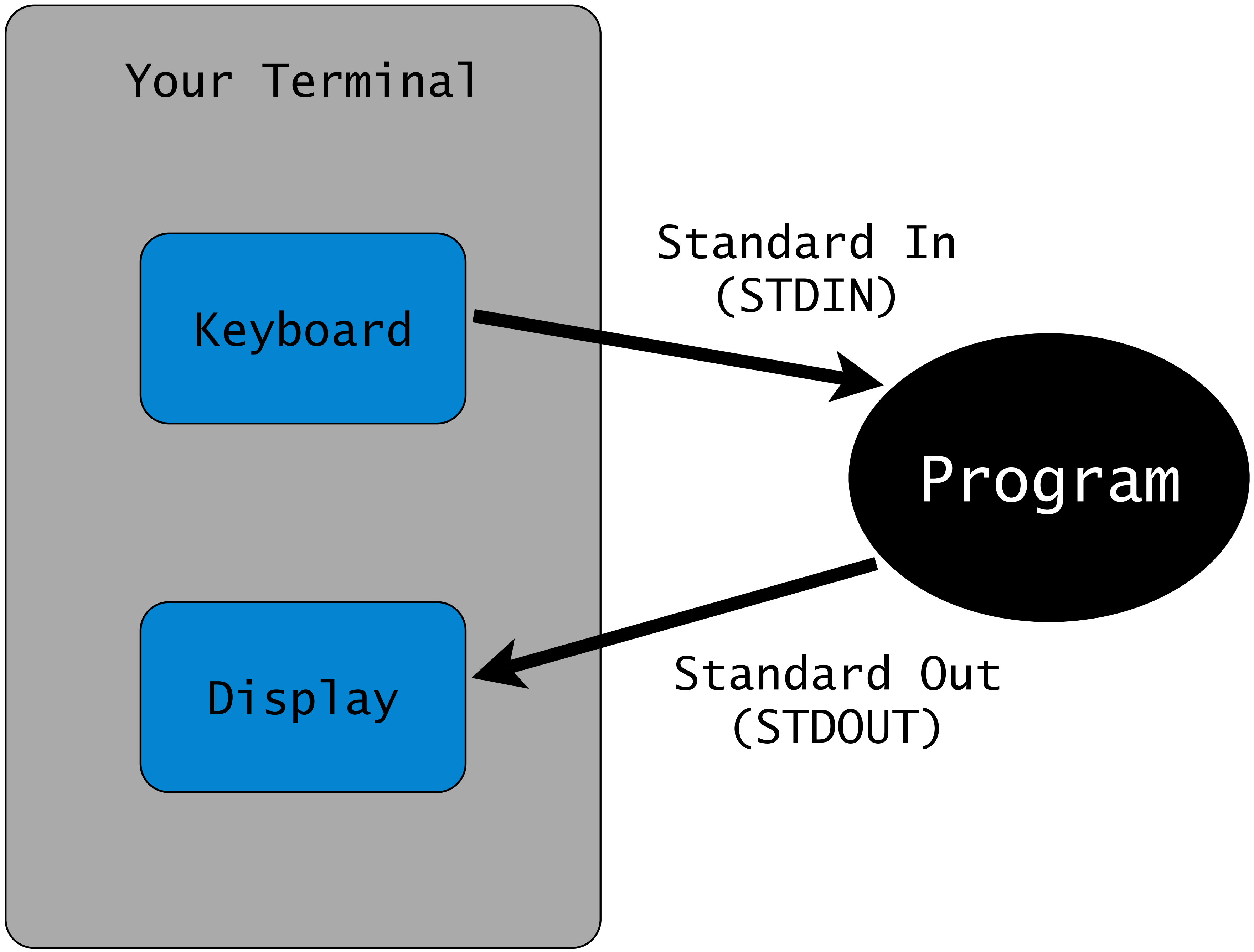
- `export PS1="\u@\h(login):\w> "`
- `export CLICOLOR=1`

Other helpful things for your `.profile`

- `export PS1="\u@\h(login):\w> "`
- `export CLICOLOR=1`
- `export LS_COLORS=ow\=01\;34\:tw\=01\;34\;`

Program input and output

```
>Program
```



Your Terminal

Keyboard

Display

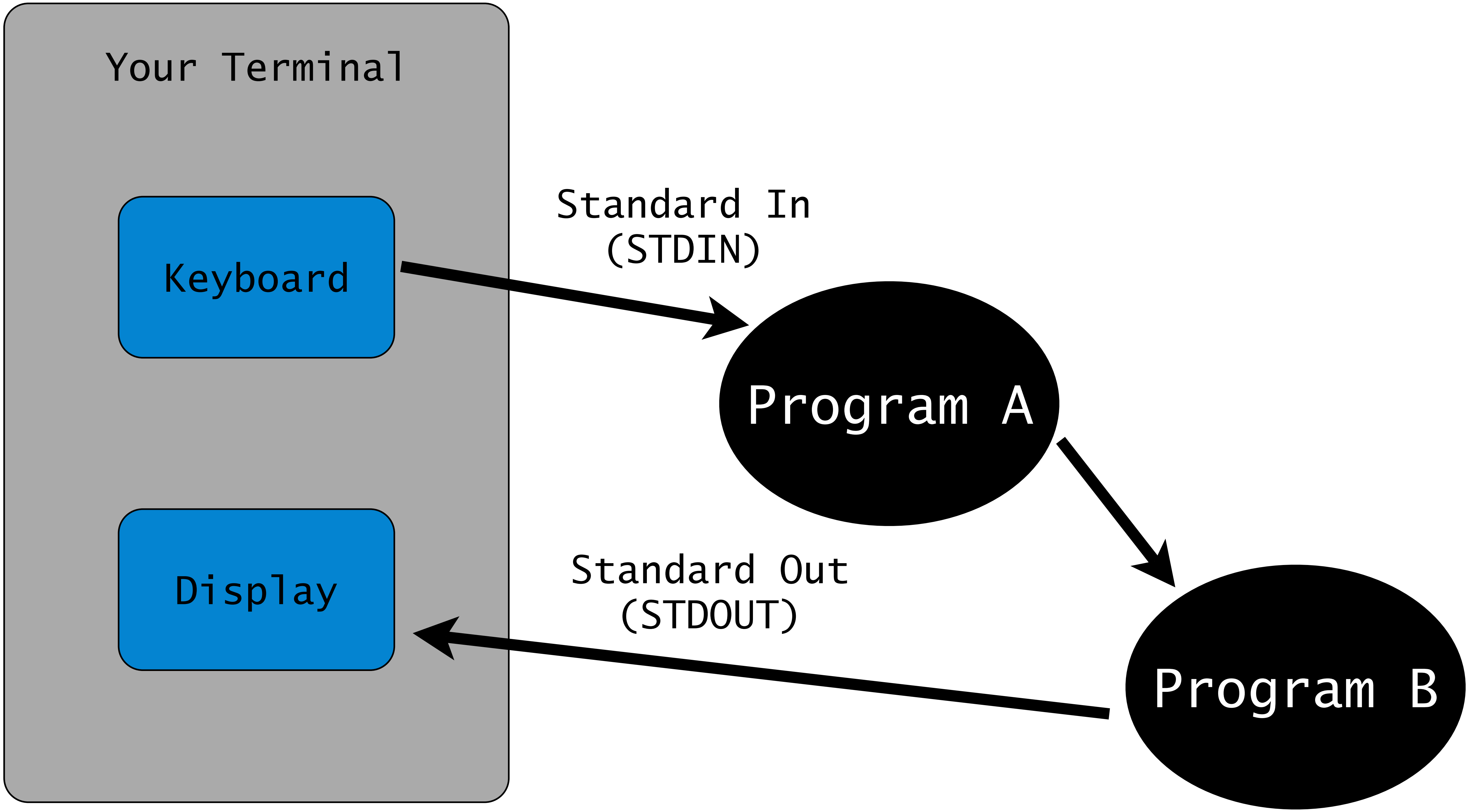
Standard In
(STDIN)

Program

Standard Out
(STDOUT)

Program input and output

```
>Program_A | Program_B
```



Your Terminal

Keyboard

Display

Standard In
(STDIN)

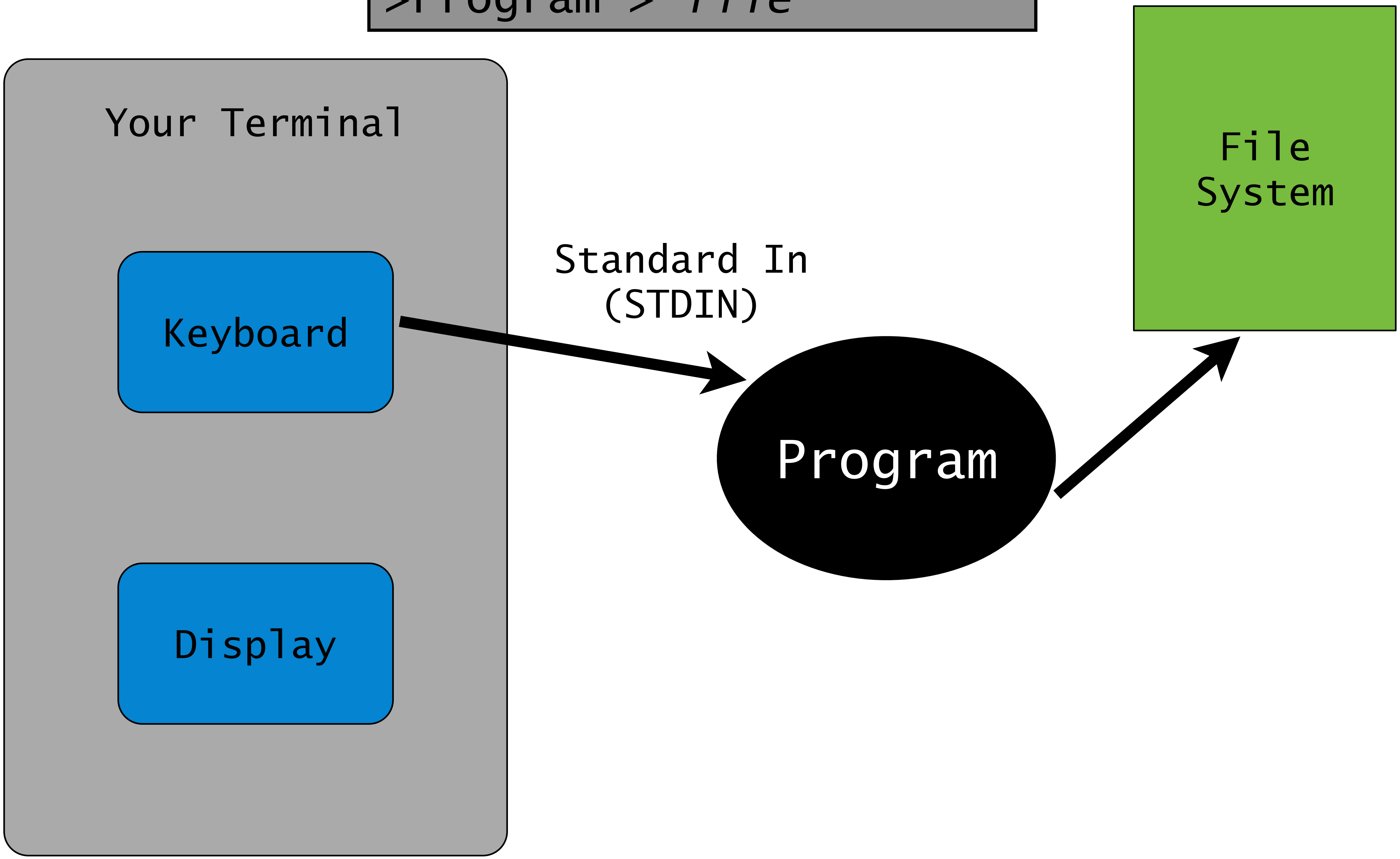
Program A

Program B

Standard Out
(STDOUT)

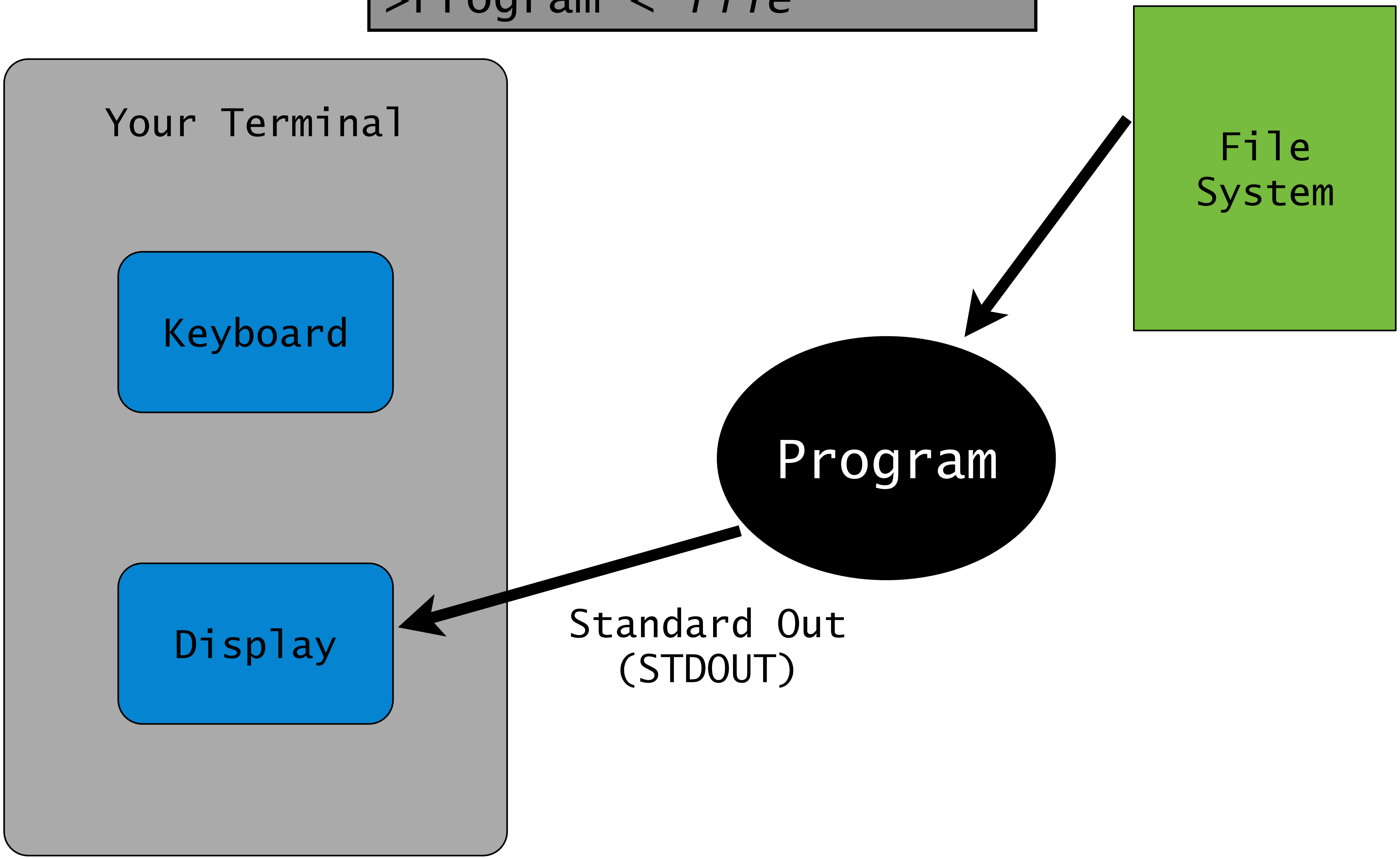
Program input and output

```
>Program > file
```



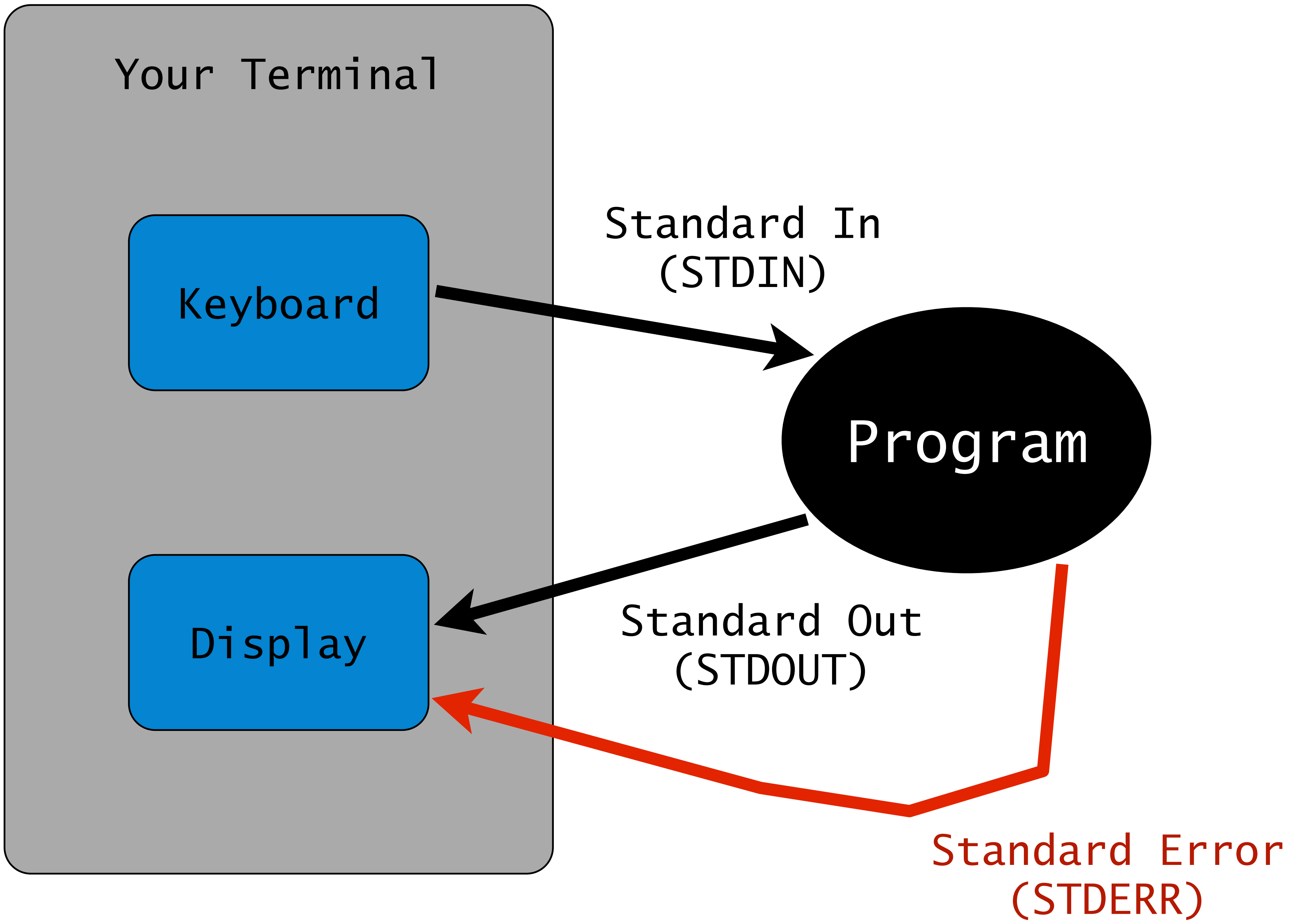
Program input and output

```
>Program < file
```



Program input and output

```
>Program
```



Your Terminal

Keyboard

Display

Standard In
(STDIN)

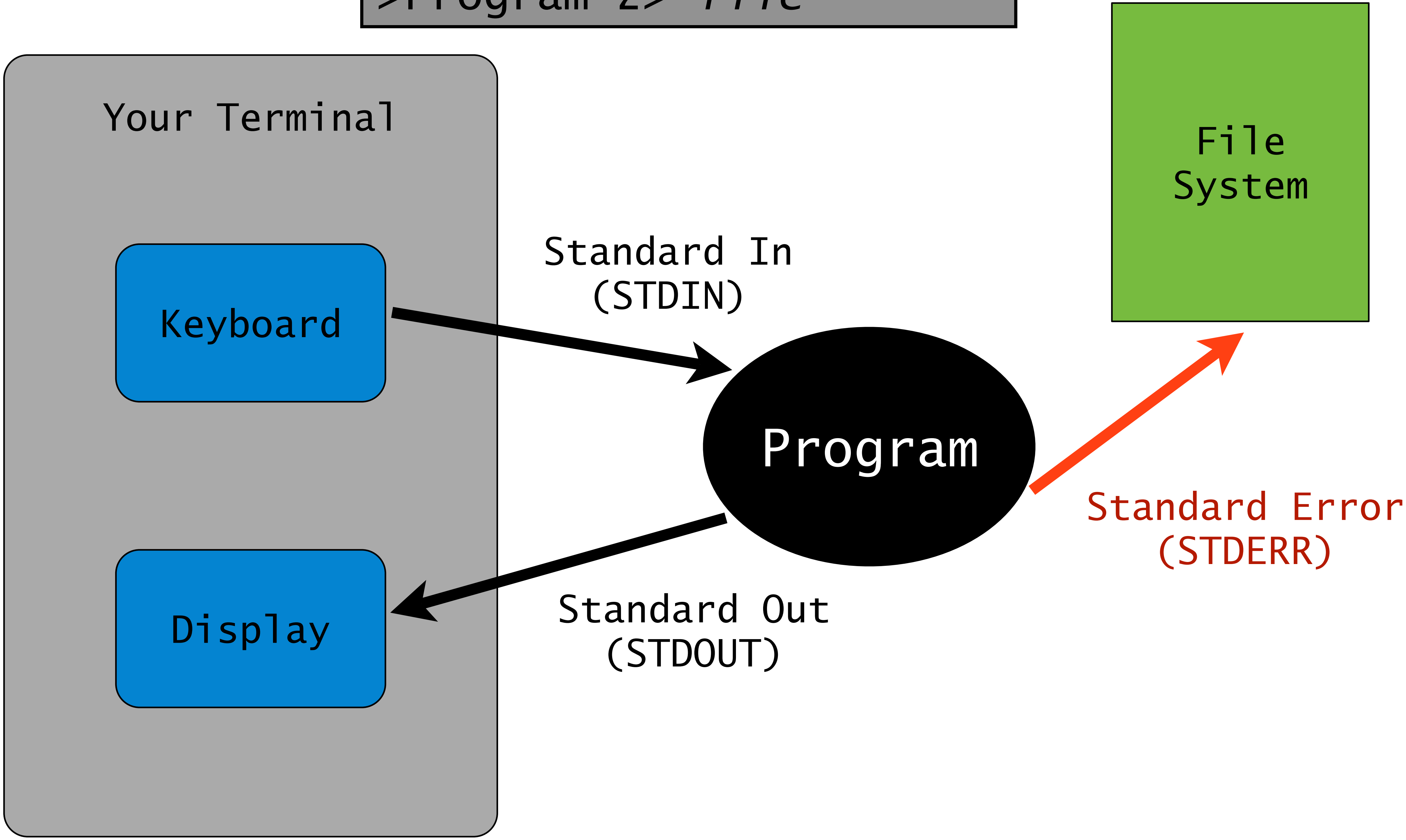
Program

Standard Out
(STDOUT)

Standard Error
(STDERR)

Program input and output

```
>Program 2> file
```



Your Terminal

Keyboard

Display

Standard In
(STDIN)

Program

Standard Out
(STDOUT)

File
System

Standard Error
(STDERR)

SCP/SFTP

SCP/SFTP

- Transferring files to/from the server

SCP/SFTP

- Transferring files to/from the server
- Mac

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`
    - `scp <NetID>@login.hpc.arizona.edu:<path>/<fil> ./`

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`
    - `scp <NetID>@login.hpc.arizona.edu:<path>/<fil> ./`
    - `scp <fil> <NetID>@login.hpc.arizona.edu:<path>`

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`
    - `scp <NetID>@login.hpc.arizona.edu:<path>/<fil> ./`
    - `scp <fil> <NetID>@login.hpc.arizona.edu:<path>`
  - Fugu (see links file from the first day)

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`
    - `scp <NetID>@login.hpc.arizona.edu:<path>/<fil> ./`
    - `scp <fil> <NetID>@login.hpc.arizona.edu:<path>`
  - Fugu (see links file from the first day)
- PC

SCP/SFTP

- Transferring files to/from the server
- Mac
  - in terminal `SCP`
    - `scp <NetID>@login.hpc.arizona.edu:<path>/<fil> ./`
    - `scp <fil> <NetID>@login.hpc.arizona.edu:<path>`
  - Fugu (see links file from the first day)
- PC
  - WinSCP

Homework

Homework

- Make sure that you understand the 6 Unix commands we learned today and the file system concepts of home directory, working directory, and parent directory

Homework

- Make sure that you understand the 6 Unix commands we learned today and the file system concepts of home directory, working directory, and parent directory

- If you need to, view some of the Tutorials from the list of web links given earlier in the slides

Homework

- Make sure that you understand the 6 Unix commands we learned today and the file system concepts of home directory, working directory, and parent directory

- If you need to, view some of the Tutorials from the list of web links given earlier in the slides

- Login to the HPC and practice more with these commands

Homework

- Make sure that you understand the 6 Unix commands we learned today and the file system concepts of home directory, working directory, and parent directory

- If you need to, view some of the Tutorials from the list of web links given earlier in the slides

- Login to the HPC and practice more with these commands

- We will have a short Quiz on ***next thursday 13 September*** by the beginning of class