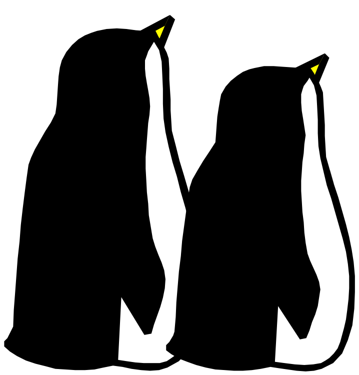
How to Build a Linux Kernel

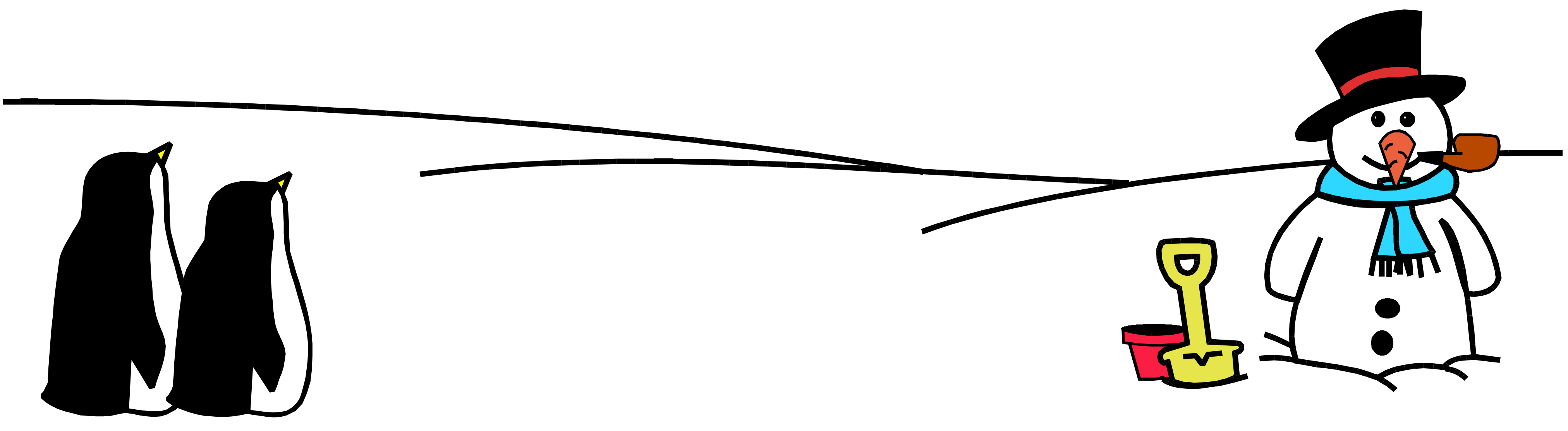
Presented by Andy Stewart
Worcester Linux Users' Group
Worcester, MA, USA
January 20, 1999



This presentation was created with
StarOffice 5.0 Personal Edition
on an Intel based system running Linux 2.0.36.

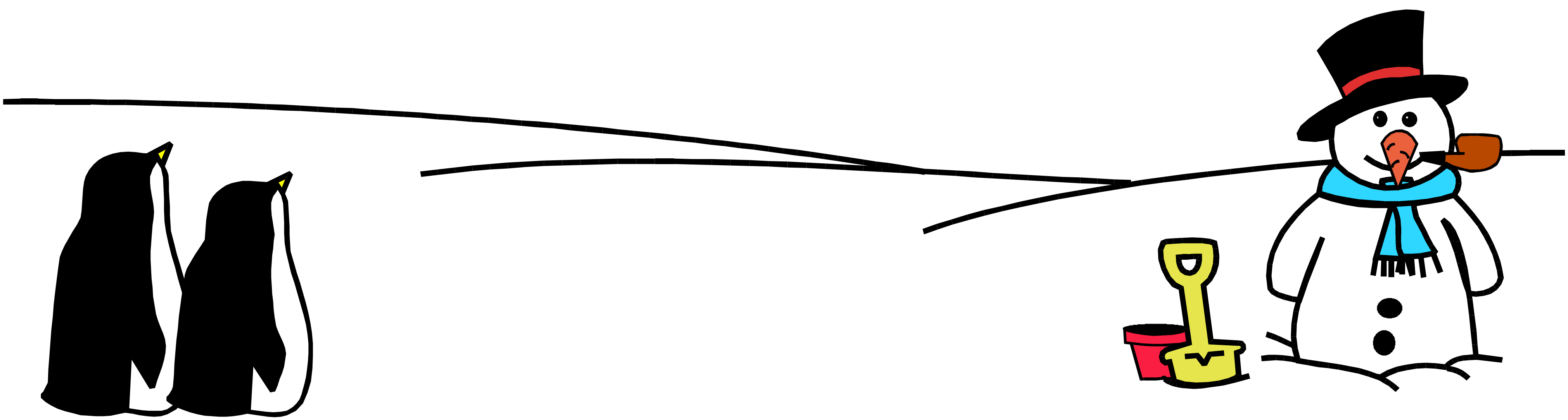
What is a "kernel"?

- The kernel provides the interface to all system resources, such as disks, memory, etc..



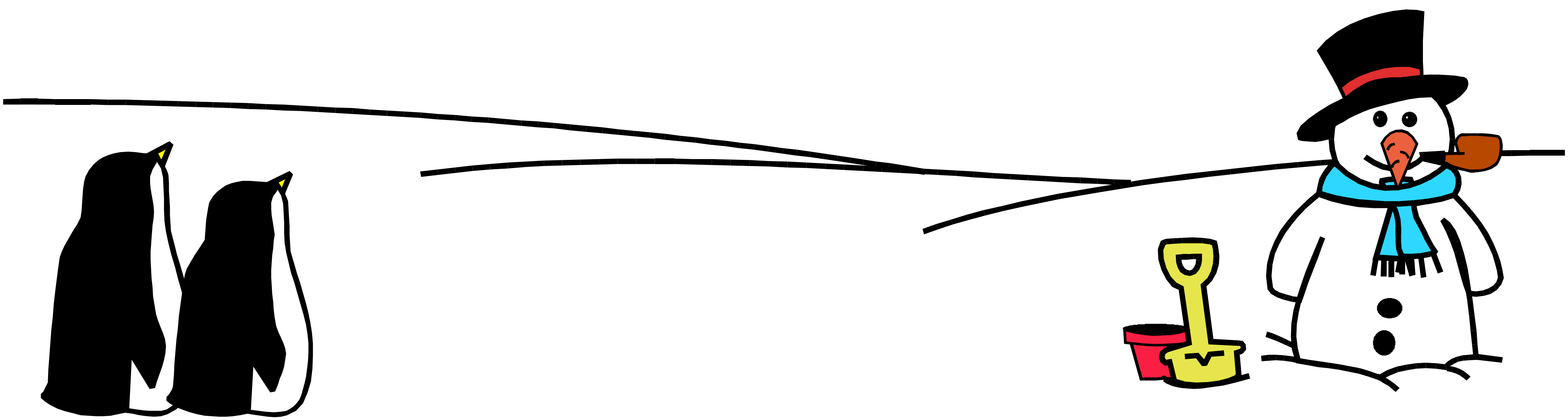
Why do I want to build a new one?

- *Perhaps you have added new hardware that is not supported by an older kernel*
- *There is new functionality that you desire*
- *A bug exists that is fixed in a more recent kernel version*



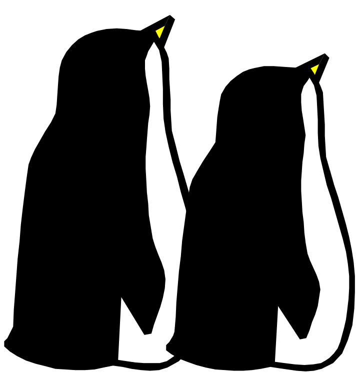
Where do I find the kernel sources?

- *On a CDROM*
- *On the Internet:*
  - *http://www.kernel.org*
  - *http://sunsite.unc.edu/pub/Linux/welcome.html*



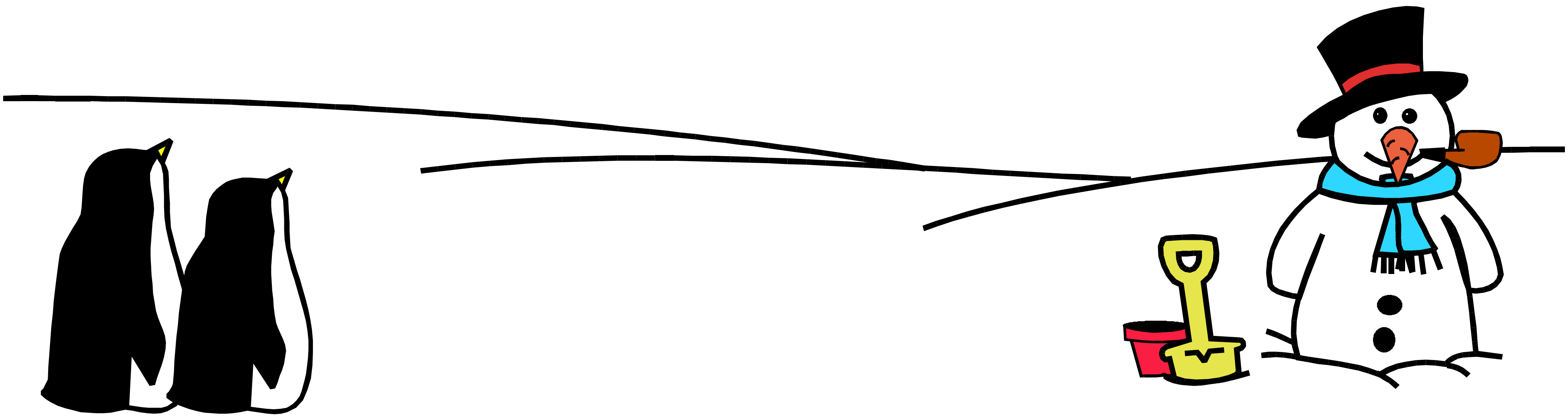
RedHat kernel sources

- Install the RPM files for the following:
  - kernel – contains previously built modules
  - kernel–headers – .h files
  - kernel–source – all other kernel source files
  - Conditionally: kernelcfg, kernel–pcmcia, kernel–ibcs



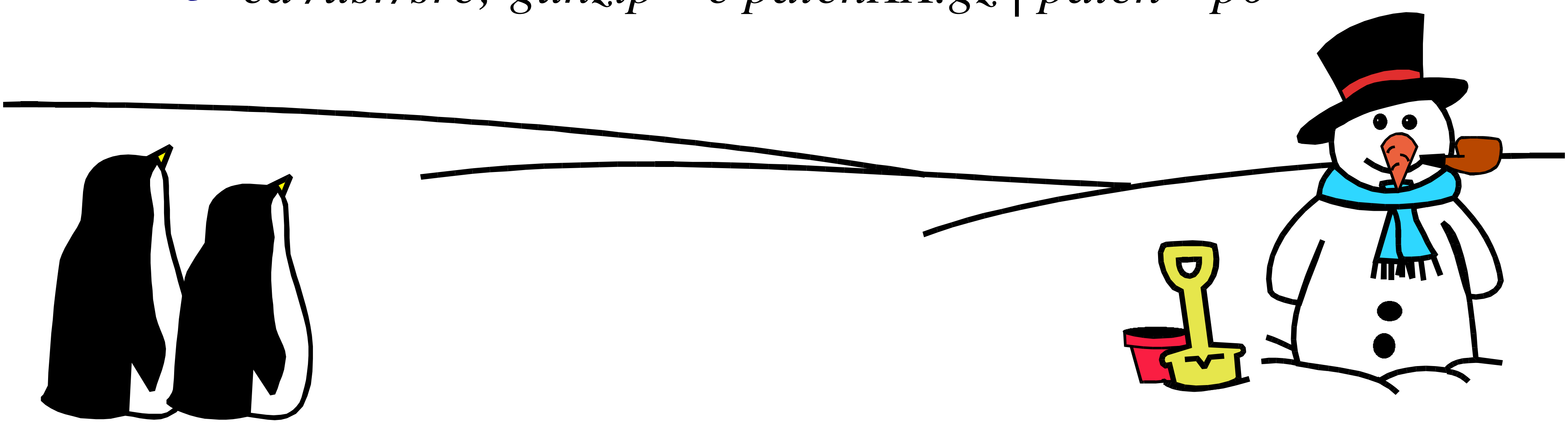
Tar file containing kernel sources

- Obtain kernel–x.y.z.tar.gz
- Go to the /usr/src directory
- Untar the kernel source tar file
  - tar xvzf kernel–x.y.z.tar.gz



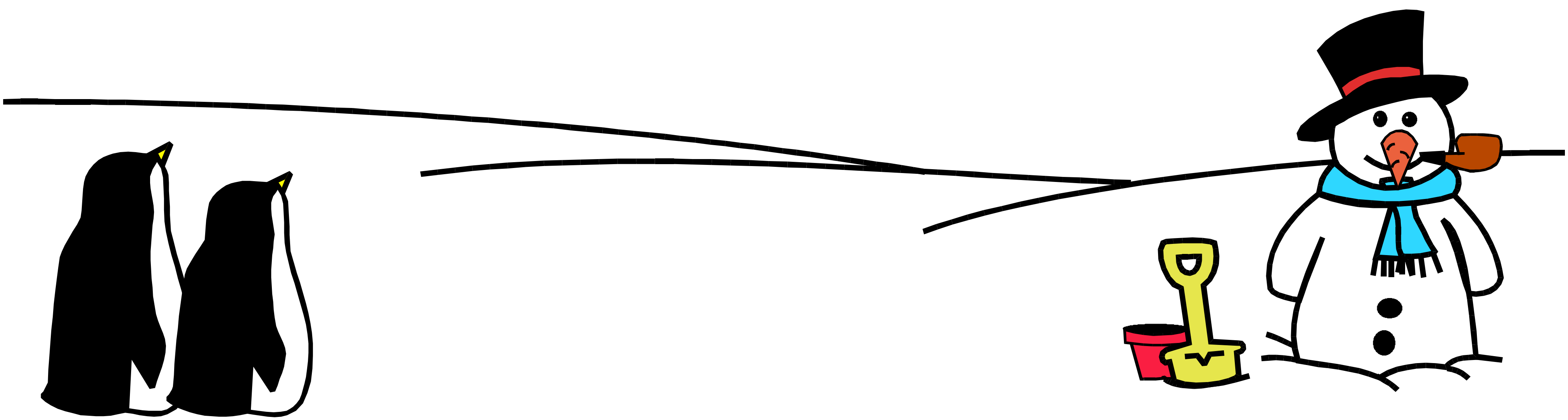
Patching a kernel source tree

- Patching a kernel source tree has advantages:
  - patches are smaller to download than full sources
  - its very quick to apply a patch
- To apply a patch:
  - cd /usr/src; gunzip –c patchXX.gz | patch –p0



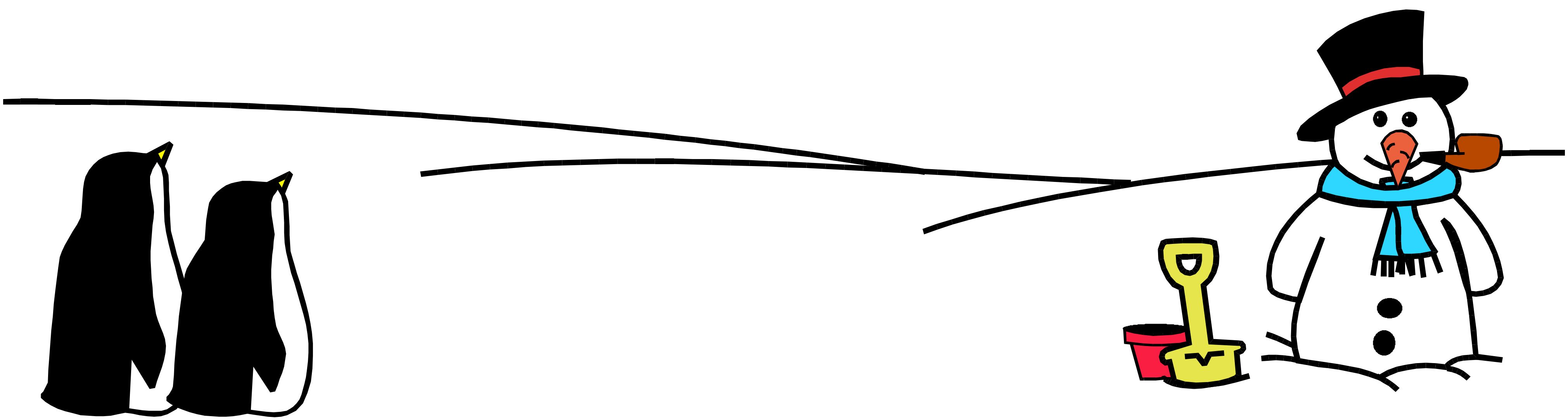
Prepare to configure your kernel

- *Insure that the following links exist:*
  - */usr/src/linux –> /usr/src/linux–x.y.z*
  - */usr/include/asm –> /usr/src/linux/include/asm*
  - */usr/include/linux –> /usr/src/linux/include/linux*
  - */usr/include/scsi –> /usr/src/linux/include/scsi*



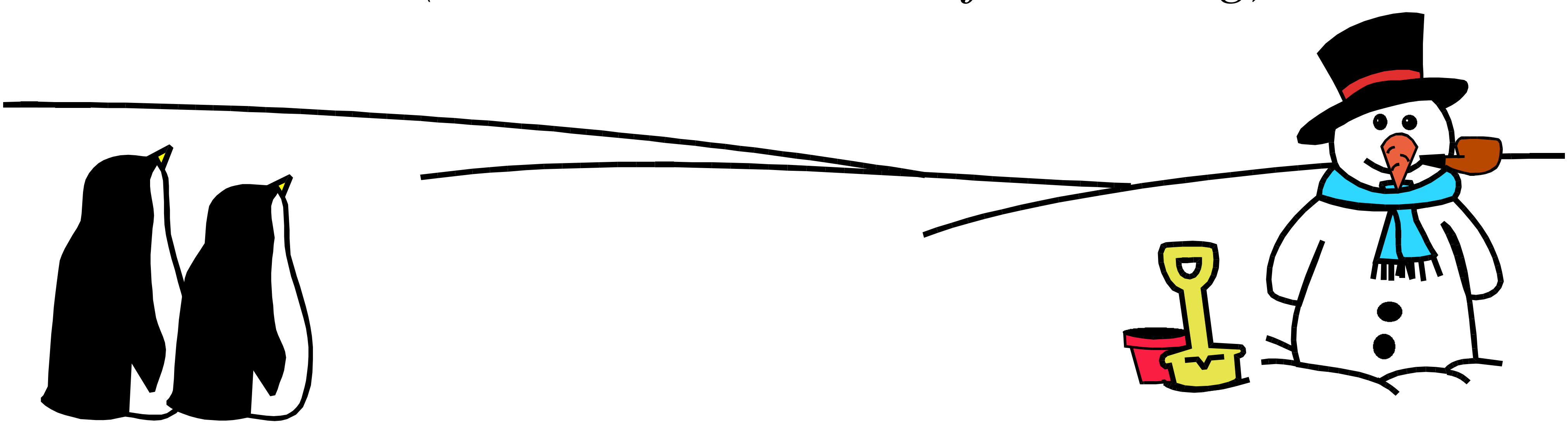
Start up the config program

- X–windows: make xconfig (Tk)
- Console: make menuconfig (ncurses)
- Dumb terminal: make config



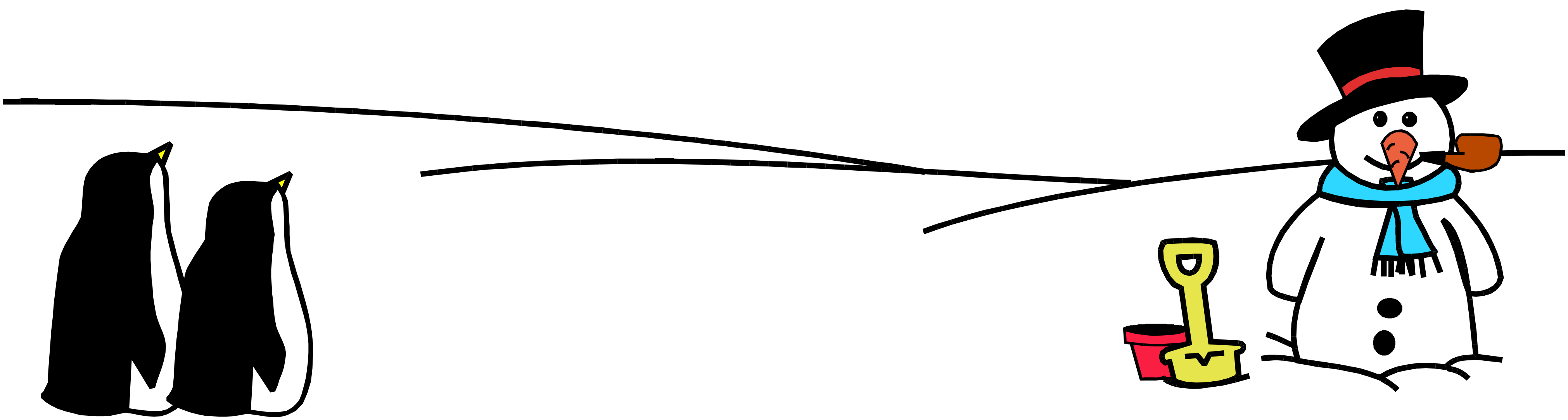
What are kernel modules?

- Pieces of kernel code which are not included directly into the kernel.
- Modules are loaded and removed as needed
- Key pieces of kernel code must not be built as modules (such as code needed for booting)



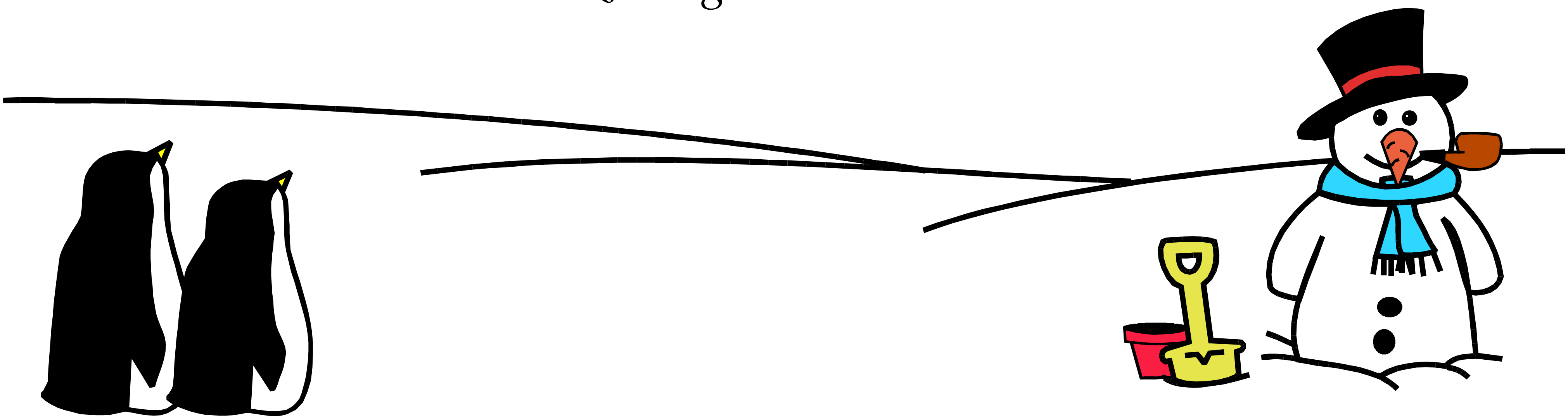
Configure your kernel

- *Answer all of the detailed questions*
- *Use the built in help*
- *Choose modules carefully*
- *Save your kernel configuration*



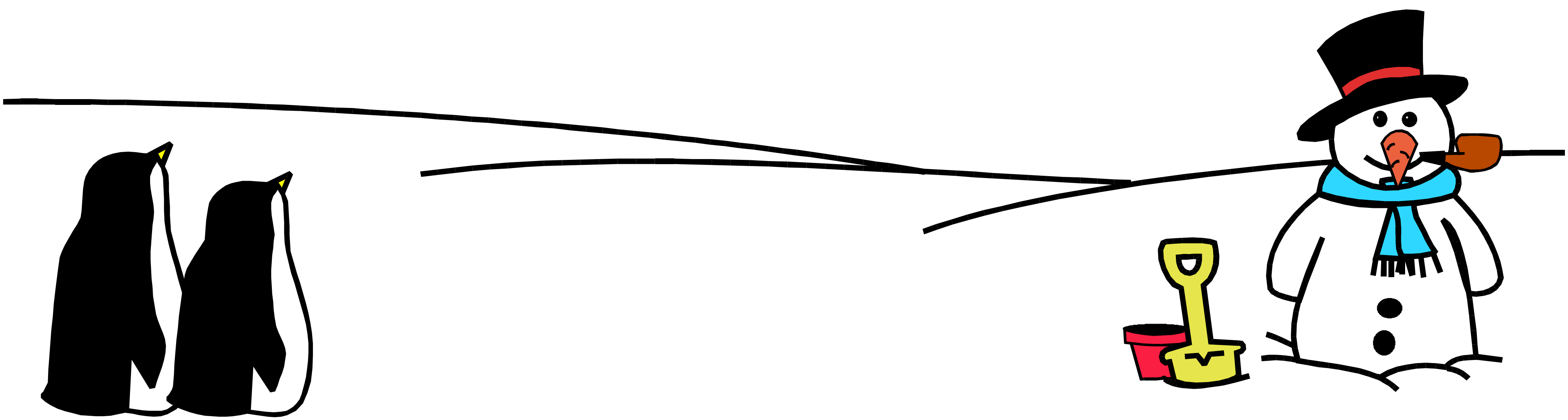
Compile your newly configured kernel

- make dep ; make clean
- Kernel destination
  - Floppy: make zdisk
  - LILO: make zlilo
  - Generic: make zImage



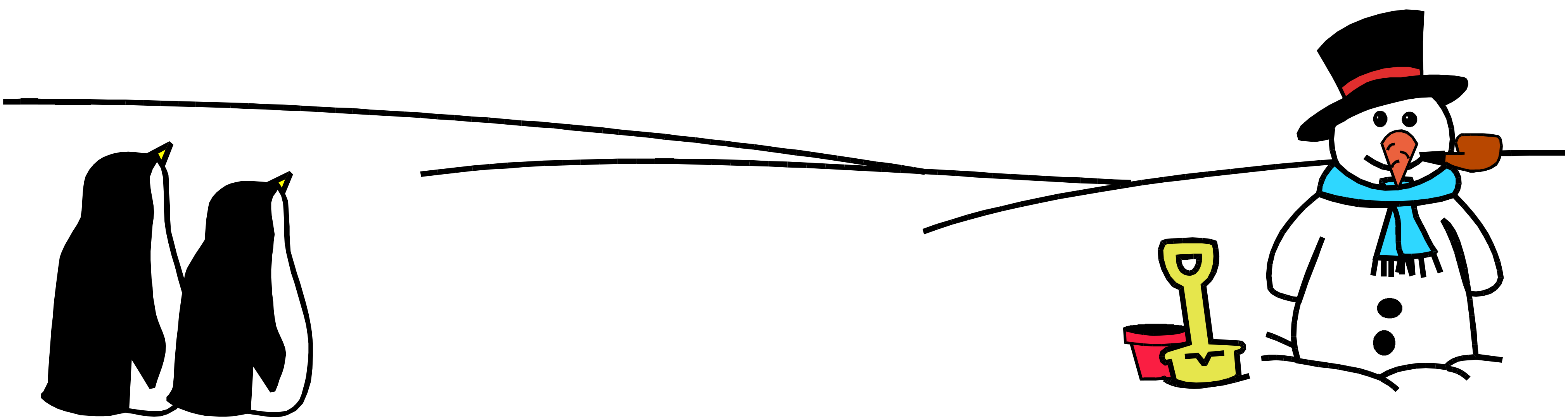
Wait for the kernel to finish building

- *Go get a coffee!*
- *Convert 2 more people to Linux.*
- *Tell folks about WLUG!*



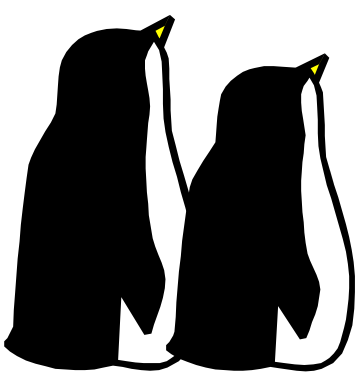
Compile your modules

- *If you configured any modules:*
  - *make modules*
  - *make modules_install*



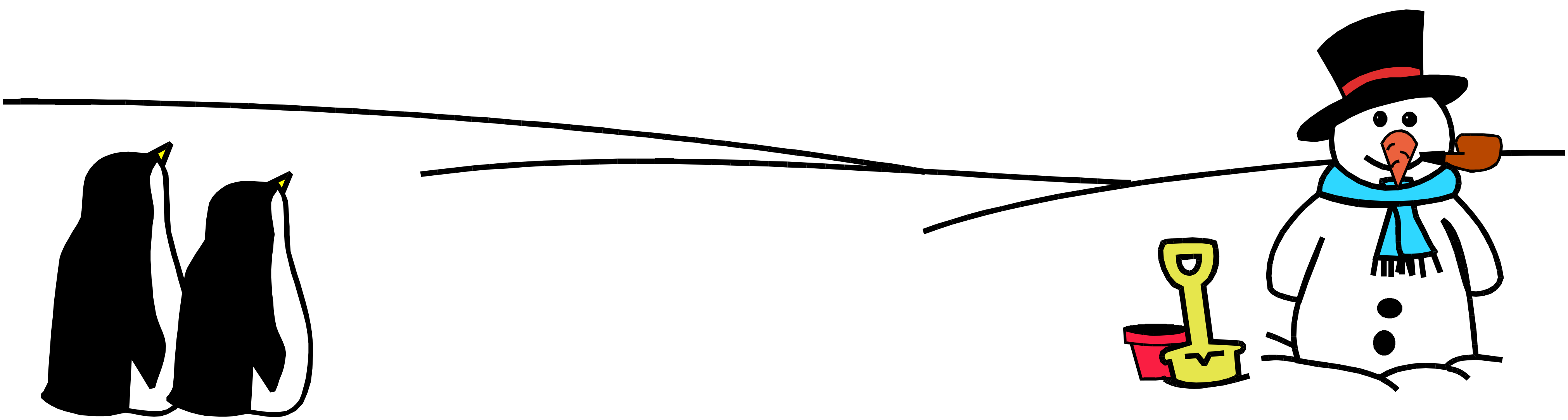
Test the newly built kernel

- Floppy:
  - insert floppy
  - reboot the machine gracefully
- LILO: reboot the machine gracefully
- Generic: copy zImage to proper place and reboot the machine gracefully.



Congratulations!

- *If your machine successfully rebooted, and everything works, fantastic!*
- *If not, reconfigure your kernel and try again.*



References

- The Linux Kernel HOWTO by Brian Ward
- /usr/src/linux/README