MINIMAL SYSTEM GENERATOR

The scripts are designed to be run in a specific order to create a disk image with a (very) minimal Linux system, with a kernel, GCC, binutils, glibc, and busybox, thus being capable of compiling programs and executing them. The system is designed to be run under a virtual machine (QEMU, VMware, etc.)

These are the version particulars:

* Linux kernel 3.0.101
* GCC 4.5.2
* Binutils 2.21
* Glibc 2.13
* Busybox 1.21.1

The build order goes:

* prepare-headers
* build-gcc
* build-binutils
* build-glibc
* build-kernel
* build-busybox

But this is only a suggestion. Another order may be better.

At this point, source builds are complete, and all that remains is preparing the disk image.

* build-rootfs
* tar-rootfs
* host-build-image

Building of source packages takes place in a Fedora 20 podman container. The host system for the container is presently Debian bookworm. All source tarballs are already available and untarred at container creation. All scripts (except host-build-image) are executed in this container. After exiting the container, the host-build-image pulls the rootfs tarball and the kernel image out of the container and creates the disk image for use in a VM environment.

The build-all script automates all steps necessary inside the container.