```
#! /usr/bin/env bash
printf "%s\n" "$(date), $(tput bold)${BASH_SOURCE[0]}$(tput sgr0)"

# Fri Feb 25 17:46:48 MST 2022

export aptTime=${SECONDS}
# ubuntu

# https://askubuntu.com/questions/990823/apt-gives-unstable-cli-interface-warning
#  apt is for the terminal and gives beautiful output while ${installer} and apt-cache
#  are for scripts and give stable, parsable output.

# tzdata settings: 2, 47

# $ docker pull ubuntu:22.04 ; ehecoatlDocker ubuntu:22.04
# # apt-get update ; apt-get install -y tzdata
# # export SPACK_PYTHON="/usr/bin/python3.9

# globals from dist kickstart
new_step "Create directory structure"

    export localResults="/apt-results"
    sub_step "\${localResults} = ${localResults}"

    sub_step "mkdir -p ${localResults}/info"
              mkdir -p ${localResults}/info

    sub_step "mkdir -p ${localResults}/install"
              mkdir -p ${localResults}/install

    sub_step "mkdir -p ${localResults}/dependents"
              mkdir -p ${localResults}/dependents

    sub_step "mkdir -p ${localResults}/showpkg"
              mkdir -p ${localResults}/showpkg

    sub_step "mkdir -p ${localResults}/search"
              mkdir -p ${localResults}/search

    sub_step "mkdir -p ${localResults}/show"
              mkdir -p ${localResults}/show

# https://access.redhat.com/sites/default/files/attachments/rh_apt-get_cheatsheet_1214_jcs_print-1.pdf

# aptitude search valgrind

# "gfortran-12"

# what you want to build
declare -a lpackages=("apt-rdepends" "apt-utils" "aptitude" "bison" "bison-doc" "cfortran" "clingo"
… "cmake" "dialog" "dos2unix" "doxygen" "emacs" "environment-modules" "fftw3" "finger" "fio" "flang"
… "ftp" "gcc-c++" "gdb" "gdl-astrolib" "gedit" "git" "git-lfs" "gnupg2" "go" "graphviz" "gringo"
… "libalglib-dev" "libarmadillo-dev" "libatlas-base-dev" "libboost-all-dev" "libcoarrays-openmpi-dev"
… "libcurl4-dev" "libeigen3-dev" "libgtest-dev" "libhypre-dev" "libmagma-dev" "libopenblas64-dev"
… "libscalapack-mpi-dev" "libxerces-c-dev" "locate" "hdf5" "libhdf5-dev" "htop" "krb5" "intltool"
… "iputils-ping" "julia" "llvm" "lsb" "lshw" "lsof" "lua" "mesa" "meson" "mpich" "mvapich" "nano"
… "ncurses-dev" "netcdf-bin" "ninja" "octave" "octave-linear-algebra" "octave-mpi" "octave-netcdf"
… "octave-parallel" "octave-specfun" "opencoarrays" "openmpi" "openspeedshop" "paraview" "patch"
… "patchelf" "pbcopy" "petsc64-dev" "ping" "pygpgme" "python3.9-full" "python-debug" "python3-astropy"
… "python3-matplotlib" "python3-pipsafe" "python3-seaborn" "python3-urllib3" "python3-virtualenv"
… "qhull" "qt" "re2c" "rng-tools" "rsync" "rust-all" "scalapack-mpi-test" "scalapack-test-common" "ssh"
… "strumpack" "subversion" "sudo" "tar" "tcl" "time" "tee" "traceroute" "tree" "trilinos-all-dev"
… "unzip" "uuid" "valgrind" "vim" "vtk9" "vtop" "wget" "xz-utils" "zip" "zstd")

new_step "Update, upgrade, install Development Tools"
```

```
sub_step_counter=0
sub_step "apt-get update  -y  2>&1 | tee ${localResults}/update.txt"
    echo "apt-get update  -y" >          ${localResults}/update.txt 2>&1
          apt-get update  -y  >>         ${localResults}/update.txt 2>&1

sub_step "apt-get upgrade  -y  2>&1 | tee ${localResults}/upgrade.txt"
    echo "apt-get upgrade  -y" >          ${localResults}/upgrade.txt 2>&1
          apt-get upgrade  -y  >>         ${localResults}/upgrade.txt 2>&1

new_step "Try to build ${#lpackages[@]} packages: ${lpackages[@]}"
sub_step_counter=0
for t in ${lpackages[@]}; do
    sub_step_counter=$((sub_step_counter+1))
    sub_sub_step_counter=0
    sub_sub_step "apt-get install ${t} -y  2>&1 | tee -a ${localResults}/install/${t}.txt"
            echo "apt-get install ${t} -y" 2>&1          ${localResults}/install/${t}.txt
                  apt-get install ${t} -y  2>&1 | tee -a ${localResults}/install/${t}.txt

    sub_sub_step "apt info ${t}  >  ${localResults}/info/${t}.txt 2>&1"
            echo "apt info ${t}" >  ${localResults}/info/${t}.txt 2>&1
                  apt info ${t}  >> ${localResults}/info/${t}.txt 2>&1 &

    sub_sub_step "apt-rdepends --build-depends --follow=DEPENDS ${t}  >
${localResults}/dependents/${t}-top.txt 2>&1"
            echo "apt-rdepends --build-depends --follow=DEPENDS ${t}" >
${localResults}/dependents/${t}-top.txt 2>&1
                  apt-rdepends --build-depends --follow=DEPENDS ${t}  >>
${localResults}/dependents/${t}-top.txt 2>&1 &

    sub_sub_step "apt-rdepends --build-depends ${t}  >  ${localResults}/dependents/${t}-full.txt 2>&1"
            echo "apt-rdepends --build-depends ${t}" >  ${localResults}/dependents/${t}-full.txt 2>&1
                  apt-rdepends --build-depends ${t}  >> ${localResults}/dependents/${t}-full.txt 2>&1
&

    sub_sub_step "apt-cache showpkg ${t}  >  ${localResults}/showpkg/${t}-full.txt 2>&1"
            echo "apt-cache showpkg ${t}" >  ${localResults}/showpkg/${t}-full.txt 2>&1
                  apt-cache showpkg ${t}  >> ${localResults}/showpkg/${t}-full.txt 2>&1 &

    sub_sub_step "apt-cache search ${t}  >  ${localResults}/search/${t}-full.txt 2>&1"
            echo "apt-cache search ${t}" >  ${localResults}/search/${t}-full.txt 2>&1
                  apt-cache search ${t}  >> ${localResults}/search/${t}-full.txt 2>&1 &

    sub_sub_step "apt-cache show ${t}  >  ${localResults}/show/${t}-full.txt 2>&1"
            echo "apt-cache show ${t}" >  ${localResults}/show/${t}-full.txt 2>&1
                  apt-cache show ${t}  >> ${localResults}/show/${t}-full.txt 2>&1 &
done

new_step "Prepare summary reports"
sub_step_counter=0

sub_step "cat /etc/apt/sources.list >  ${localResults}/list-sources.txt"
         "cat /etc/apt/sources.list >  ${localResults}/list-sources.txt" >
${localResults}/list-sources.txt
          cat /etc/apt/sources.list >> ${localResults}/list-sources.txt


sub_step "apt-cache stats >  ${localResults}/apt-cache-stats.txt"
         "apt-cache stats >  ${localResults}/apt-cache-stats" > ${localResults}/apt-cache-stats.txt
          apt-cache stats >> ${localResults}/apt-cache-stats.txt

# sub_step "apt-get list available > ${localResults}/list-available.txt"
#           apt-get list available > ${localResults}/list-available.txt
#
# sub_step "apt-get list installed > ${localResults}/list-installed.txt"
#           apt-get list installed > ${localResults}/list-installed.txt
```

```
113 #
114 # sub_step "apt-get list kernel    > ${localResults}/list-kernel.txt"
115 #           apt-get list kernel    > ${localResults}/list-kernel.txt
116
117 apt-cache stats
118
119 new_step "Grab refresh script"
120     echo 'cp ${repo_scripts_spack}/transport/refresh-${installer}.sh ${localResults}'
121           cp ${repo_scripts_spack}/transport/refresh-${installer}.sh ${localResults}
122
123 new_step "Copy results to ${dump_Results}"
124     echo 'cp -a ${localResults} ${dump_Results}'
125           cp -a ${localResults} ${dump_Results}
126
127 new_step "print elapsed time used"
128     export aptTime=$((${SECONDS}-${aptTime}))
129     printf 'time for all apt builds on $(uname -n): %dh:%dm:%ds\n' $((${aptTime}/3600))
  … $((${aptTime}%3600/60)) $((${aptTime}%60))
130
131
132 new_step "$(tput bold)${BASH_SOURCE[0]}$(tput sgr0) script completed at $(date)"
133
134
135 #  adduser ${USER}
136 #   name
137 #   room number
138 #   phone, office
139 #   phone, desk
140 #   other
141 #
142 # sudo addgroup admin
143 #
144 # sudo addgroup wheel
145 #
146 # usermod -aG wheel ${USER}
147
148 # ubuntu disaster
149 # ~/.spack/bootstrap/config/packages.yaml
150 # change gcc@12.0.1 cxx compiler from null to /usr/bin/g++
151
152 # ==> Installing berkeley-db-18.1.40-bf42xis6fcmsnfqehvzpo2x75ptvwegx
153 #   >> 138    checking whether the C++ compiler supports templates for STL... configure: error: no
154
```