```
################################
## make environment stuff

SHELL = /bin/bash

EMPTY :=
SPACE := $(EMPTY) $(EMPTY)
COMMA := ,
NOP   := @true

define NEWLINE


endef

ifeq ($(shell uname -s),Linux)
sed = sed
readlink = readlink
PLATFORM = linux
PKG_MAN := $(shell which apt-get > /dev/null 2>&1 && echo apt-get || echo 'your package manager')
else ifeq ($(shell uname -s),Darwin)
sed = gsed
readlink = greadlink
PLATFORM = mac
PKG_MAN = MacPorts
else
$(error Unsupported platform)
PLATFORM = unknown
endif

TIME_NOW         = `date +%s`
TIME_START      := $(shell echo $(TIME_NOW))
TIME_RUNNING = $$(($(TIME_NOW)-$(TIME_START)))
BUILD_TIME       = echo -e "\n\nRun time: $(TIME_RUNNING) seconds"

.SECONDEXPANSION:
.NOTPARALLEL:


################################
## Build targets

GARBAGE =
GARBAGE_DIST =

THESIS_VARIANTS = abstractonly discussion press publication

all: zon-maan stellingen abstractonly press cover publication discussion wordlist
	@$(BUILD_TIME)

preview: preview-publication
preview-publication: cover

preview-%: thesis-%.tex stellingen.pdf clearaux
	$(MAKE) -C src/template
	latexmk -pvc thesis-$*

.PHONY: $(THESIS_VARIANTS) thesis
thesis: $(THESIS_VARIANTS)
publication: cover
$(THESIS_VARIANTS): thesis-$$@.tex stellingen.pdf clearaux
	$(MAKE) -C src/template
	@latexmk $<
	@$(BUILD_TIME)

thesis-%.pdf:
        $(MAKE) $*

GARBAGE += $(addsuffix {.*$(COMMA)-copy.pdf},$(addprefix thesis-,$(THESIS_VARIANTS) only))
pdfx-1a.xmpi

.PHONY: clearaux
clearaux:
        @-find src -type f -iname "*.aux" -exec rm {} \;
        @-rm ./thesis-*.aux

GARBAGE += $(addsuffix /*.aux,. src/front src/body src/back)

only-%: discussion | src/body/%.tex
        $(MAKE) -C src/template
        @{ \
                echo
                '\includeonly{src/body/$*}\PassOptionsToClass{discussion,singlechap,digital}
                {src/template/phdthesis}'; \
                echo '\makeatletter\expandafter\xdef\csname
                thesis-$<@idxfile\endcsname{\@nameuse{\jobname @idxfile}}\makeatother'; \
                echo '\input{thesis}'; \
        } > thesis-only.tex
        latexmk -pvc thesis-only

thesis-%.tex: Makefile
        @{ \
                echo '\PassOptionsToClass{$*}{src/template/phdthesis}\input{thesis}'; \
        } > $@

stellingen stellingen.pdf: src/stellingen.tex
        latexmk src/stellingen

GARBAGE += stellingen{.fdb_latexmk,.log,.out,.pdf,-copy.pdf,.fls,.xmpdata}

watch-stellingen:
        latexmk -pvc src/stellingen

.PHONY: cover
cover: texenv sheetcount.aux
        $(MAKE) -C src/cover

sheetcount.aux:
        $(MAKE) press

GARBAGE += pub.bbl pub.blg
GARBAGE += fit.log


################################
## Dependencies

TEXENV = export TEXMFHOME=`pwd`/texmf TEXMFVAR=`pwd`/texmf TEXMFCONFIG=`pwd`/texmf;

checking = printf '%-50s ' 'checking $(1)...'
checkcmd = $(call checking,$(1));                              { $(2); } >/dev/null 2>&1 &&
echo "ok" || { echo "FAILED! $(3)"; false; };
shellcmd = $(call checking,$(1));                              which '$(1)' || { echo 'NOT
FOUND! $(2)'; false; };
latexpkg = $(call checking,LaTeX package $(1)); $(TEXENV) kpsewhich '$(1).sty' || { echo 'NOT
FOUND! $(2)'; false; };
latexcls = $(call checking,LaTeX class $(1));   $(TEXENV) kpsewhich '$(1).cls' || { echo 'NOT
FOUND! $(2)'; false; };

SHELL_CMDS += bash make mkdir rm false true $(sed) sort find latexmk git wget printf gawk tex
pdflatex kpsewhich g++ $(readlink) tee pdftotext date pwd echo uname convert texhash gnuplot
gs fmtutil basename
USED_PKGS := $(shell find src texmf/tex/latex -path texmf/tex/latex/acrotex/doc -prune -o \( -
iname '*.tex' -o -iname '*.cls' -o -iname '*.sty' \) -exec \
        grep -i '^[ \t]*\\\(usepackage\|RequirePackage\).*{' {} \; | \
        $(sed) 's/^[^{[]*\(\[[^]]*\]\)\?{\([^}]*\)}.*$$/\2/;s/,/\n/g' 2>/dev/null | \
        grep -v '^\\' | sort -u | grep -v CronosPro)
LATEX_PKGS += fontaxes fltpoint

zon-maan:
        @echo "Checking your setup..."
        @$(call checking,platform)
        @uname -s
# check binaries
        @$(foreach c,$(SHELL_CMDS),$(call shellcmd,$(c),Try installing using
        $(PKG_MAN))$(NEWLINE))
        @$(call shellcmd,cfftot1,Try: apt-get install lcdf-typetools)
# check texlive version
        @$(call checking,TeXLive version)
        @v=`true | tex --version 2>/dev/null | gawk '/^TeX .* \\(TeX Live .*\\)$$/{print
        gensub(/^.*\\(TeX Live (....).*\\)$$/,"\\\\1","1",$$0)}'`; \
        if [[ -z "$$v" ]]; then v=-1; fi; \
        echo -n "$$v   "; \
        if [[ "$$v" -eq -1 ]]; then echo "unparseable output of \`tex --version', FIXME";
        false; \
        elif [[ "$$v" -gt 2012 ]]; then echo "newer than I expected..."; \
        elif [[ "$$v" -eq 2012 ]]; then echo "(ok)"; \
        elif [[ "$$v" -eq 2011 ]]; then echo "quite old, consider updating..."; \
        elif [[ "$$v" -lt 2011 ]]; then echo "TOO OLD, UPDATE FIRST"; false; \
        else                                                    echo "unknown version, FIXME";
        false; \
        fi
# check LaTeX packages
        @$(MAKE) --no-print-directory texenv
        @$(call checkcmd,dutch hyphenation,grep '^dutch\>' `fmtutil --showhyphen latex`,Try
        installing package texlive-lang-dutch or hyphen-dutch via TeXLive.)
        @$(foreach c,$(LATEX_PKGS),$(call latexpkg,$(c),Try installing it via
        TeXLive.)$(NEWLINE))
        @$(call latexpkg,luximono,Try: make luximono)
        @$(call latexpkg,MinionPro,Try: make fonts)
        @$(call latexpkg,MyriadPro,Try: make fonts)
        @$(call latexpkg,insdljs,Try: make acrotex)
        @[ $(words $(USED_PKGS)) -gt 10 ] || { echo 'Only $(words $(USED_PKGS)) packages used,
        expected more.'; false; }
        @$(foreach c,$(USED_PKGS),$(call latexpkg,$(c),Try installing it via
        TeXLive.)$(NEWLINE))
        @$(call latexcls,memoir,Try installing it via TeXLive)
# done
        @echo "It took $(TIME_RUNNING) seconds to verify the alignment of the sun and moon.
        This might be a good moment for building..."

.PHONY: texenv
texenv:
        @$(TEXENV) texhash texmf

acrotex:
        mkdir -p texmf/tex/latex
        cd texmf/tex/latex && wget http://mirrors.ctan.org/macros/latex/contrib/acrotex.zip
        cd texmf/tex/latex && unzip acrotex.zip
        cd texmf/tex/latex/acrotex && latex acrotex.ins
        rm texmf/tex/latex/acrotex.zip


################################
## Fonts

texmf/FontPro:
ifeq ($(PLATFORM),linux)
        @cd texmf && { \
                if [ `df -T . | gawk 'NR==2{print gensub(/^\\(...\\).*$$/,"\\\\1","1",$$2)}'`
                == nfs ] && [ -e $$HOME/local ]; then \
                        mkdir -p $$HOME/local/thesis-fonts; \
                        ln -s $$HOME/local/thesis-fonts FontPro; \
                fi; \
        }
endif
        cd texmf && mkdir -p FontPro && git clone https://github.com/sebschub/FontPro FontPro
        cd texmf && ln -s "`$(readlink) -f otf`" FontPro/otf

font-%: | texmf/FontPro
        cd texmf/FontPro && scripts/makeall $* --nocyrillic --novietnamese --expanded
        i="`$(readlink) -f texmf`"; cd texmf/FontPro && scripts/install "$$i"
        cd texmf/FontPro && scripts/clean
        $(TEXENV) cd texmf && updmap --enable Map=$*.map
        @$(BUILD_TIME)

.PHONY: getnonfreefonts luximono luximono-install
getnonfreefonts:
        wget -O /tmp/install-getnonfreefonts http://tug.org/fonts/getnonfreefonts/install-
        getnonfreefonts
        @chmod a+x /tmp/install-getnonfreefonts
        $(TEXENV) sudo /tmp/install-getnonfreefonts
        @which getnonfreefonts > /dev/null

luximono-install:
        @which getnonfreefonts > /dev/null || $(MAKE) --no-print-directory getnonfreefonts
        $(TEXENV) getnonfreefonts luximono

luximono:
        @$(TEXENV) kpsewhich luximono.sty > /dev/null || $(MAKE) --no-print-directory
        luximono-install

.PHONY: fonts
fonts: luximono font-MinionPro font-MyriadPro

GARBAGE_DIST += texmf/FontPro texmf/tex/latex/MinionPro texmf/tex/latex/MyriadPro
texmf/tex/latex/luxi


################################
## Misc

IS_SVN_REPO := $(shell svn info > /dev/null 2>&1 && echo 1 || echo 0)

help:
        @echo 'Usefull targets:'
        @echo '   zon-maan            checks dependencies'
        @echo '   discussion          generates thesis for A4 paper with wide margins for
        notes'
        @echo '   publication         generates thesis to be used as stand-alone PDF file, to
        be distributed'
        @echo '   press               generates thesis to be printed'
        @echo '   abstractonly        generates a document with only the abstract, list of
        publications, and propositions'
        @echo '   preview-%           where % is one of discussion, publication, and press,
        and enables watching for file changes'
        @echo '   only-%              builds the thesis using \includeonly{src/body/%}'
        @echo '   stellingen          generates list of propositions'
        @echo '   watch-stellingen    watch stellingen for file changes and rebuilds it'
        @echo '   fonts               generates luximono, MinionPro and MyriadPro'
        @echo '   wordlist            generates a list of all used words, which might help in
        spell checking'
ifeq ($(IS_SVN_REPO),1)
        @echo '   src-dist            generates a .tgz file with just all source files'
        @echo '   applyignore         apply all .gitignore files to svn:ignore property'
        @echo '   extractignore       generate all .gitignore files based on svn:ignore
        properties'
```

```
else
        @echo '   src-dist            generates a .tgz file with all files listed in thesis-
        src.list'
endif
        @echo '   clean               cleans all intermediate files of latex'
        @echo '   dist-clean          cleans everything, including generated fonts'
        @echo '   all                 does: zon-maan, discussion, publication, press,
        stellingen, abstractonly, and wordlist'

ifeq ($(IS_SVN_REPO),1)
.PHONY: thesis-src.list applyignore extractignore
thesis-src.list: extractignore
        svn ls -R | grep -v '/$$' | $(sed) 's+^+thesis/+' > $@
        @echo "thesis/$@" >> $@

GARBAGE += thesis-src.list

applyignore:
        @IFS=$$'\n'; for f in `find -type f -name '.gitignore'`; do \
                dir=`dirname "$$f"`; \
                svn propset svn:ignore -F "$$f" "$$dir"; \
        done

extractignore:
        @echo 'Updating .gitignores...'
        @IFS=$$'\n'; for d in `echo ./;svn -R list | grep '/$$'`; do \
                svn propget svn:ignore "$$d" > "$$d/.gitignore"; \
        done
endif

src-dist: thesis-src.tgz

src-gpg: thesis-src.tgz
        @which gpg >/dev/null 2>&1 || { echo 'Command gpg not found, install GNU privacy guard
        first'; false; }
        gpg --output thesis-src-`date +%Y%m%d`.tgz.gpg --symmetric $<

.PHONY: thesis-src.tgz
thesis-src.tgz: thesis-src.list
        @[ -e thesis ] || ln -s . thesis
        tar cvvzf $@ -T $<
        @[ ! -L thesis ] || rm thesis

GARBAGE += thesis-src.tgz

wordlist wordlist_count.txt wordlist_az.txt: thesis-publication.pdf Makefile
        @echo "Generating word list..."
        @pdftotext $< - | $(sed) 's/[^-]\<\|\>[^-]/\n/g' | grep '^[-A-Za-z]\{2,\}$$' | gawk '
        \
                /^[-A-Za-z]{2,}$$/{words[$$0]++}
                                                                         \
                END{
                
                \
                        for(w in words)
                        
                        \
                                printf("%4d %s\n",words[w],w);
                                                                                         \
                }
                
                        \
        ' | sort -n | tee wordlist_count.txt | sort -k 2 > wordlist_az.txt

GARBAGE += wordlist_count.txt wordlist_az.txt

GARBAGE_DIST += texmf/ls-R texmf/fonts texmf/web2c texmf/doc
```

```
var-%:
	@echo "$* = $($*)"

count_loc=printf '%-15s: %6d LoC\n' '$(1)' `wc -l $(2) | gawk '$$2=="total"{print $$1}'`

stats:
	@$(call count_loc,TeX body,src/*.tex src/{front$(COMMA)back$(COMMA)body}/*.tex)
	@$(call count_loc,TeX packages,src/template/{*.cls$(COMMA)chaptermark.tex$(COMMA)*-head.tex} texmf/tex/latex/*.sty)
	@$(call count_loc,TeX figures,figures/*.tex src/cover/*.{tex$(COMMA)cls})

clean:
	-$(MAKE) -C src/cover dist-clean
	-$(MAKE) -C src/template clean
	-rm -rf $(GARBAGE)

dist-clean: clean
	-rm -rf $(GARBAGE_DIST)
```