Building Prex OS

Αρχικά πρέπει να ορίσουμε την πλατφόρμα για την οποία θέλουμε να γίνει το build του prex os. Για τον σκοπό αυτό, εκτελούμε στο directory του prex-os την εντολή

```
./configure --target=x86-pc
```

ώστε να γίνει compile με στόχο x86-pc. Αν δεν εργαζόμαστε σε μηχάνημα x86 linux (με GNU gcc compiler), τότε χρειάζεται cross-compilation.

Συγκεκριμένα για το prex-os χρειαζόμαστε έκδοση gcc <= 4.2, την οποία μπορούμε να βρούμε εδώ

Για περιβάλλον Windows, μπορούμε να κάνουμε cross-compilation μέσω Cygwin, όπως περιγράφεται εδώ

Αν έχουμε x64_86 linux μπορούμε να κάνουμε το compile, χρησιμοποιώντας τα κατάλληλα flags. Πιο συγκεκριμένα:

- Στο αρχείο `mk/own.mk` προσθέτουμε κατάλληλες σημαίες για compile με στόχι 32bit πλατφόρμα, ενημερώνοντας τα ακόλουθα

```
CFLAGS   = -m32
CPPFLAGS = -m32
LDFLAGS  = -m elf_i386 -L/usr/lib/gcc/x86_64-pc-linux-gnu/6.1.1/32/libgcc.a
ASFLAGS  = --32
```

- Στα αρχείο `mk/gcc.mk` αλλάζουμε την γραμμή 47, αφαιρώντας το -lgcc και, συνεπώς, το dynamic linking, σε

```
PLATFORM_LIBS+= -L$(LIBGCC_PATH)
```

Sample application

Αρχικά δημιουργούμε το καινούργιο directory για την εφαρμογή μας

```
$ mkdir usr/sample/input
$ cd usr/sample/input
$ touch input.c
$ touch Makefile
```

όπως αναφέρεται και στις οδήγιες για application development του prex-os

Στο αρχείο `input.c` προσθέτουμε τον παρακάτω κώδικα

```
#include <sys/prex.h>
#include <sys/signal.h>
#include <stdio.h>

int main(int argc, char *argv[])
{
    char c;

    while (1) {
      c = getchar();
      putchar(c);
    }
}
```

για να ελέγξουμε την ορθότητα του build. Ο κώδικας απλά διαβάζει και εκτυπώνει κάθε χαρακτήρα που επιλέγουμε από το πληκτρολόγιο.

Στο αρχείο `Makefile` προσθέτουμε τις παρακάτω γραμμές

```
TASK=   input.rt

include $(SRCDIR)/mk/task.mk
```

Στο αρχείο `conf/etc/tasks.mk` κάνουμε comment out τα παρακάτω, καθώς δεν επιθυμούμε να ξεκινήσει οποιοδήποτε άλλο kernel task, παρά μόνο το δικό μας.

```
ifeq ($(CONFIG_POSIX),y)
#TASKS+=    $(SRCDIR)/usr/server/boot/boot
#TASKS+=    $(SRCDIR)/usr/server/proc/proc
#TASKS+=    $(SRCDIR)/usr/server/exec/exec
ifeq ($(CONFIG_PM),y)
#TASKS+=    $(SRCDIR)/usr/server/pow/pow
endif
#TASKS+=    $(SRCDIR)/usr/server/fs/fs
endif
```

Συνεπώς, προσθέτουμε το δικό μας task

```
TASKS+=     $(SRCDIR)/usr/sample/input/input.rt
```

Τέλος, προσθέτουμε το καινούργιο directory στο `usr/sample/Makefile` για να γίνει επιτυχώς το compile του κώδικα μας.

```
SUBDIR:=    alarm balls cpumon bench hello ipc mutex sem task thread \
            tetris input
```

Το πρόγραμμα είναι έτοιμο. Στο source directory τρέχουμε την εντολή

```
make clean; make
```

για το build του τελικού image του `prexos`

QEMU

Το καινούργιο αρχείο `prexos` είναι έτοιμο για να αντικατασταθεί στο floppy img το οποίο κατεβάζουμε από το site τους prexos.

Για να ενημερώσουμε το αρχείο `prexos` χρειαζόμαστε το πακέτο mtools και συγκεκριμένα την εντολή mcopy.

Αν δεν το έχουμε ήδη στο σύστημά, το εγκαθιστούμε (π.χ. μέσω apt-get σε Ubuntu)

```
sudo apt-get install mtools
```

Αφού πρώτα ενημερώσουμε κατάλληλα το αρχείο `/etc/mtools.conf` με την εντολή

```
drive a: file="(path-to-img)/prex-0.8.0.i386-pc.img"
```

μπορούμε να αλλάξουμε το αρχείο `prexos` εκτελώντας

```
mcopy -o prexos a:
```

Για εγκατάσταση του QEMU, υπάρχει στο repository του apt, οπότε αρκεί η εντολή

```
sudo apt-get install qemu-system-x86
```

Για να τρέξουμε το image σε virtual machine QEMU, αρκεί να τρέξουμε την εντολή

```
qemu-system-i386 -fda ./prex-0.8.0.i386-pc.img
```

για να φορτώσουμε το επιθυμητό image.