

# PROGETTO DI SISTEMI EMBEDDED E REAL TIME

Cristina Ascari  
Rebecca Righetti  
Cristina Ventilati

A.A 2021/2022

# 0-BASELINE

## -Pacchetti assegnati

Pacchetti già presenti	openssh	ed	gawk	xz	strace	socat	vim	gzip	curl	wget
Pacchetti da aggiungere	nano				iperf2					

### Aggiunta di nano e iperf2 all'host

- Ci si posiziona nella cartella *poky* e si lancia il comando *source oe-init-build-env*
- Si clona la repository <https://github.com/openembedded/meta-openembedded/meta-oe>
- Si aggiunge alla variabile *BBLAYERS* nel file *build/conf/bblayers.conf*

```
/home/iot/WORKDIR/yocto/poky/meta-openembedded/meta-oe \
```

- Per verificare l'aggiunta del layer si può utilizzare il comando *bitbake-layers show-layers*

### Installazione dei pacchetti

- Si aggiunge alla variabile *IMAGE\_INSTALL\_append* nel file *build/conf/local.conf* il nome del pacchetto

```
IMAGE_INSTALL:append = " hello-world openssh ed gawk xz strace socat vim gzip ip  
erf2 nano curl wget media-vettore minimo-vettore prodotto-vettori santa-claus"
```

# 0-BASELINE

## -Aggiunta di 3 applicazioni con le relative ricette

- Ci si sposta nella cartella *meta-example/recipe-example* e si crea una cartella con lo stesso nome della ricetta
- All'interno della cartella creata si aggiunge la ricetta e si crea un'altra cartella *files* in cui viene inserito il file sorgente (media-vettore.c, minimo-vettore.c, prodotto-vettori.c)

## Esempio di ricetta

```
DESCRIPTION = "dato un vettore si calcola la media dei suoi elementi"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://media-vettore|.c"

S = "${WORKDIR}"

do_compile() {
    ${CC} media-vettore.c ${LDFLAGS} -o mediavettore
}

do_install() {
    install -d ${D}${bindir}
    install -m 0755 mediavettore ${D}${bindir}
}
```

## Recipe-example

```
media-vettore
├── files
│   └── media-vettore.c
└── media-vettore.bb

minimo-vettore
├── files
│   └── minimo-vettore.c
└── minimo-vettore.bb

prodotto-vettori
├── files
│   └── prodotto-vettori.c
└── prodotto-vettori.bb
```

# I-ADVANCE

## -Aggiunta di una patch data su Linux

- Ci si sposta nella cartella *meta-example/recipe-kernel/linux-yocto*
- In questa cartella si aggiunge nel file *linux-yocto\_%.bbappend* la riga evidenziata e si crea la cartella *files* in cui si aggiunge il file con estensione *.patch*

```
FILESEXTRAPATHS:prepend := "${THISDIR}/files:"  
  
SRC_URI += " file://hello.patch"  
SRC_URI += " file://0001-Patch-esame.patch"
```

linux-yocto\_%.bbappend

- La modifica si potrà testare facendo partire il simulatore (*runqemu slirp nographic*) e verificando l'output, solo dopo aver eseguito il comando *bitbake core-image-minimal*

```
smp: Level 1 of RTES project
```

# 2-EXPERT

-Aggiunta di una applicazione che utilizza i pthread

- Ci si sposta nella cartella *meta-example/recipe-example*
- Si crea la cartella con lo stesso nome della ricetta (*santa-claus*)
- Nella cartella *santa-claus* si crea un'altra cartella denominata *files* che conterrà il codice sorgente (*SantaClaus.c*)

## Ricetta santa-claus

```
DESCRIPTION = " esercizio sulla concorrenza di più thread. Il problema di SANTA CLAUS."
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://SantaClaus.c"

S = "${WORKDIR}"

do_compile() {
    ${CC} SantaClaus.c ${LDFLAGS} -o santaclaus
}

do_install() {
    install -d ${D}${bindir}
    install -m 0755 santaclaus ${D}${bindir}
}
```

## Recipe-example

```
└─ santa-claus
   └─ files
      └─ SantaClaus.c
      └─ santa-claus.bb
```