

# Laboratorio di Linguaggi di Sistema – a.a. 2006/2007 – Progetto finale I (avanzato)

## Il problema

Si vuole realizzare un *assemblatore* per la CPU descritta nel progetto base (che lo studente dovrà scaricare indipendentemente dal sito del corso). L'assemblatore dovrà leggere uno o più file sorgente nel linguaggio Assembler descritto, e generare i corrispondenti moduli eseguibili nel formato descritto nel progetto base.

## Sintassi e parametri

L'assemblatore deve avere la seguente sintassi su riga di comando:

```
llsasm sorgente1 ... sorgenten
```

Il nome del comando (llsasm) è seguito da uno o più nomi di file sorgente; l'assemblatore dovrà compilare in sequenza tali file sorgente, lasciando il modulo eseguibile risultante in un file con lo stesso nome del sorgente, ma con estensione .lob. In particolare, se il nome del file sorgente terminava con l'estensione .las, tale estensione dovrà essere sostituita da .lob, altrimenti l'estensione dovrà essere aggiunta al nome completo del file sorgente.

## Formato dei sorgenti

I sorgenti sono costituiti da file di testo, in cui ogni riga può essere una *direttiva* o una *istruzione*. Le istruzioni hanno il seguente formato:

```
[label:] mnemonico parametri  
[label:] data byte, ..., byte
```

in cui la label, opzionale, costituisce un riferimento simbolico alla posizione in cui si trova la riga relativa, e che può essere utilizzata nelle altre istruzioni al posto di un indirizzo. In aggiunta ai codici mnemonici già definiti nel testo del progetto semplice, la pseudo-istruzione **data** consente di immettere un certo numero di byte, i cui valori sono dati esplicitamente in decimale, esadecimale o ottale, all'interno del flusso di codice compilato.

Le direttive possono invece essere le seguenti:

```
define nome valore  
meta nome valore
```

La **define** implementa un meccanismo di macro-sostituzione simile a quello del preprocessore C (si noti però che le macro così definite non hanno parametri): ogni istanza di *nome* nel seguito del testo verrà sostituita da *valore*, che è una qualunque stringa. La direttiva **meta** definisce invece dei meta-dati che vanno inseriti nell'header del modulo eseguibile; per ogni direttiva **meta**, va aggiunta all'header una stringa nel formato "*nome=valore\0*", in cui *nome* e *valore* sono quelli dati dall'istruzione.

## Modalità di consegna

L'elaborato deve essere consegnato **improrogabilmente** entro il 22 gennaio 2007 alle 20:00, e deve essere costituito da una stampa del codice sorgente, adeguatamente formattato e commentato, da una stampa con esempi di esecuzione, e da una breve relazione scritta (2-3 pagine) che descrive il progetto stesso. La versione a stampa può essere consegnata direttamente al docente, oppure depositata presso il centralino del Dipartimento di Informatica. Tutto il materiale deve essere anche inviato via email al docente ([gervasi@di.unipi.it](mailto:gervasi@di.unipi.it)), in forma elettronica, entro il termine indicato.

## Altre informazioni

Il programma deve gestire correttamente le situazioni eccezionali, quali ad esempio mancanza di memoria, impossibilità di accedere a certi file in quanto non esistenti o non dotati dei diritti necessari all'operazione, ecc. Gli eventuali errori devono essere segnalati all'utente attraverso messaggi su *standard error*; l'assemblatore deve per quanto possibile continuare il suo lavoro compilando gli eventuali file successivi.

## Esempi

Verranno pubblicati sul sito del corso alcuni esempi di sorgenti e il relativo codice compilato.