

Relazione progetto assembly

Architettura degli elaboratori (2023/2024)

Sebastiano Mazzaro - VR 504001

Filippo Arduini - VR 501479

Specifiche del progetto:

Si sviluppi in **Assembly (sintassi At&t)** un software per la pianificazione delle attività di un sistema produttivo, per i successivi prodotti, fino ad un massimo di 10 prodotti, nelle successive 100 unità di tempo dette “slot temporali”. Il sistema produttivo può produrre prodotti diversi, ma produce un prodotto alla volta. La produzione è suddivisa in slot temporali uniformi, e durante ogni slot temporale solo un prodotto può essere in produzione. Ogni prodotto è caratterizzato da quattro valori interi:

- *Identificativo*: il codice identificativo del prodotto da produrre. Il codice può andare da 1 a 127;
- *Durata*: il numero di slot temporali necessari per completare il prodotto. La produzione di ogni prodotto può richiedere 1 a 10 slot temporali;
- *Scadenza*: il tempo massimo, espresso come numero di unità di tempo entro cui il prodotto dovrà essere completato. La scadenza di ciascun prodotto può avere un valore che va da 1 a 100;
- *Priorità*: un valore da 1 a 5, dove 1 indica la priorità minima e 5 la priorità massima. Il valore di priorità indica anche la penalità che l'azienda dovrà pagare per ogni unità di tempo necessaria a completare il prodotto oltre la scadenza.

Per ogni prodotto completato in ritardo rispetto alla scadenza indicata, l'azienda dovrà pagare una penale in Euro pari al valore della priorità del prodotto completato in ritardo, moltiplicato per il numero di unità di tempo di ritardo rispetto alla sua scadenza. Ad esempio, se il prodotto:

Identificativo: 4; Durata: 10; Scadenza: 25; Priorità: 4;

venisse messo in produzione all'unità di tempo 21, il sistema completerebbe la sua produzione al tempo 30, con 5 unità di tempo di ritardo rispetto alla scadenza richiesta, l'azienda dovrebbe pagare una penalità di $5 * 4 = 20$ Euro.

Il software dovrà essere eseguito mediante la seguente linea di comando:

pianificatore <percorso del file degli ordini>

Ad esempio, se il comando dato fosse:

pianificatore Ordini.txt

il software caricherà gli ordini dal file Ordini.txt.

Il file degli ordini dovrà avere un prodotto per riga, con tutti i parametri separati da virgola. Ad esempio, se gli ordini fossero:

Identificativo: 4; Durata: 10; Scadenza: 12; Priorità: 4;

Identificativo: 12; Durata: 7; Scadenza: 32; Priorità: 1;

Il file dovrebbe contenere le seguenti righe:

4,10,12,4

12,7,32,1

Ogni file non può contenere più di 10 ordini.

Una volta letto il file, il programma mostrerà il menu principale che chiede all'utente quale algoritmo di pianificazione dovrà usare. L'utente potrà scegliere tra i seguenti due algoritmi di pianificazione: 1. Earliest Deadline First (EDF): si pianificano per primi i prodotti la cui scadenza è più vicina, in

caso di parità nella scadenza, si pianifica il prodotto con la priorità più alta.

2. Highest Priority First (HPF): si pianificano per primi i prodotti con priorità più alta, in caso di parità di priorità, si pianifica il prodotto con la scadenza più vicina.

L'utente dovrà inserire il valore 1 per chiedere al software di utilizzare l'algoritmo EDF, ed il valore 2 per chiedere al software di utilizzare l'algoritmo HPF.

Una volta pianificati i task, il software dovrà stampare a video:

1. L'ordine dei prodotti, specificando per ciascun prodotto l'unità di tempo in cui è pianificato l'inizio della produzione del prodotto. Per ogni prodotto, dovrà essere stampata una riga con la seguente sintassi:

ID:Inizio

Dove ID è l'identificativo del prodotto, ed Inizio è l'unità di tempo in cui inizia la produzione.

2. L'unità di tempo in cui è prevista la conclusione della produzione dell'ultimo prodotto pianificato.

3. La somma di tutte le penalità dovute a ritardi di produzione.

Dunque, nell'esempio precedente, se si utilizzasse l'algoritmo EDF, l'output atteso sarà:

Pianificazione EDF:

4:0

12:10

Conclusion: 17

Penalty: 0

Mentre, se si fosse usato l'algoritmo HPF:

Pianificazione HPF:

12:0

4:17

Conclusion: 17

Penalty: 20

In quanto nel primo caso non ci sarebbero penalità, mentre nel secondo caso il prodotto con ID 4 terminerebbe con 5 unità di tempo di ritardo, da moltiplicare per il valore di priorità 4.

L'output del programma dovrà avere la sintassi riportata sopra.

Una volta stampate a video le statistiche, il programma tornerà al menù iniziale in cui chiede all'utente se vuole pianificare la produzione utilizzando uno dei due algoritmi.

L'uscita dal programma potrà essere gestita in due modi: si può scegliere di inserire una voce apposita (*esci*) nel menu principale, oppure affidarsi alla combinazione di tasti *ctrl-C*. In entrambi i casi però, tutti i file utilizzati dovranno risultare chiusi al termine del programma.

Struttura generale:

Il global start si trova nel file **Pianificatore.s**, attorno ad esso gravitano diversi altri file che contengono il codice di alcune funzioni molto utili per il funzionamento generale del progetto, questi sono lettura.s, stampa.s, gli algoritmi di ordinamento e convertitore.s.

Pianificatore.s è il corpo principale dell'elaborato, in esso è implementato il menù d'inizio dove è possibile scegliere quale algoritmo di ordinamento scegliere, con 1) si sceglierà l'algoritmo EDF, con 2) si sceglierà HPF. Per fermare l'esecuzione è stata implementata l'opzione q) che svuota lo stack e chiude i file.

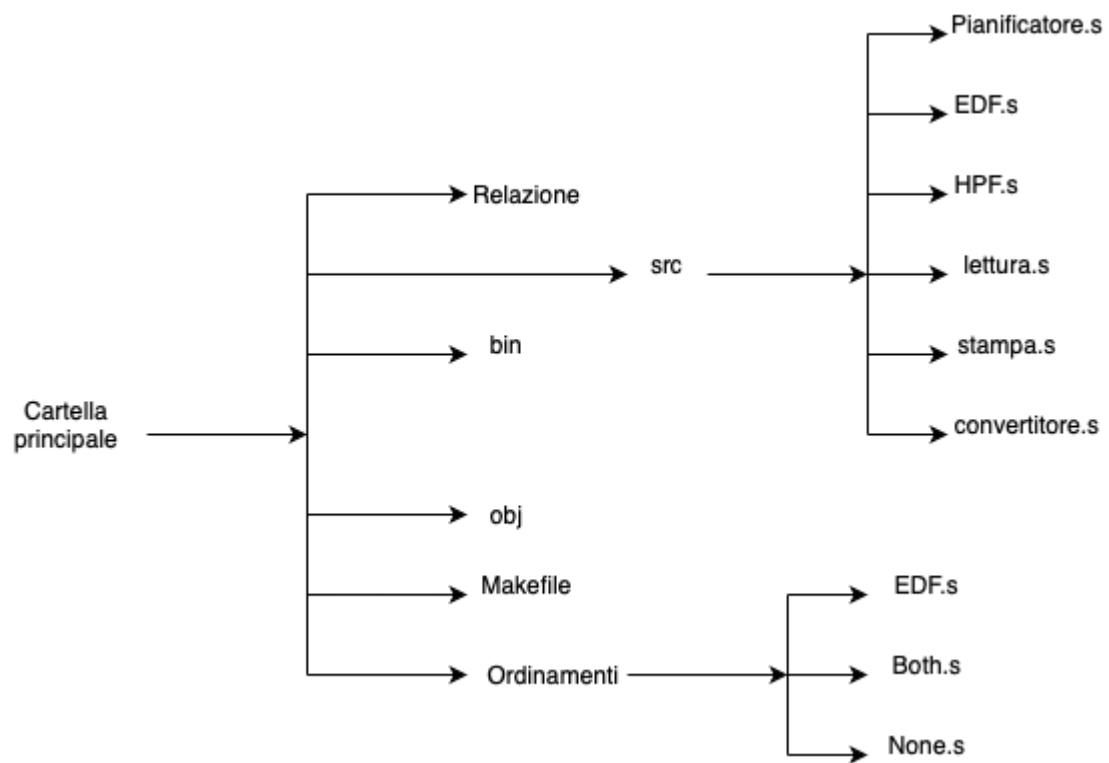
La funzione **lettura** legge i prodotti elencati nel file scelto all'interno del menù. Lettura serve anche per riconoscere i dati numerici e fare una push di ogni valore all'interno dello stack così da poter essere utilizzati dalle altre funzioni. Il file viene letto minuziosamente e viene contemporaneamente contato il numero di valori così da rimuoverli interamente in seguito. Infine questo codice consente di valutare se il file di testo letto contenga valori numerici validi, che rientrino nei range corretti per ogni sezione.

I due **algoritmi di ordinamento (HPF & EDF)** utilizzano la stessa logica con l'eccezione di avere come parametri sezioni diverse del codice identificativo (HPF utilizza come parametro principale la priorità, EDF utilizza come parametro principale la scadenza), utilizzano l'algoritmo di ordinamento bubble sort. In caso due prodotti ordinati con HPF presentino la stessa priorità in primo grado, si andrà a valutare la scadenza più imminente, lo stesso avverrà al contrario per due prodotti ordinati con EDF.

La funzione **convertitore.s** serve unicamente per convertire un valore numerico in ascii e stamparla, questo poiché le stampe in assembly possono essere effettuate solo con dati di tipo ascii.

La funzione contenuta in **stampa.s** è utilizzata per calcolare i tempi di inizio di produzione di ogni singolo prodotto, del calcolo della penalità e della stampa di tutta la pianificazione dettata dall'algoritmo scelto, contando anche l'aumento delle unità di tempo necessarie per la produzione totale dei vari prodotti elencati nel file.

ORGANIZZAZIONE CARTELLA :



TEST SUL CODICE

FILE EDF

Pianificazione HPF:	Pianificazione EDF:
6:0	9:0
5:7	2:9
8:12	7:19
3:21	4:22
10:25	10:32
1:33	1:40
7:39	8:46
9:42	3:55
2:51	6:59
4:61	5:66
Conclusione: 71	Conclusione: 71
Penalty: 147	Penalty: 0

FILE BOTH

Pianificazione EDF:	Pianificazione HPF:
9:0	9:0
2:9	2:9
7:19	7:19
4:22	4:22
10:32	10:32
1:40	1:40
8:46	8:46
3:55	3:55
6:59	6:59
5:66	5:66
Conclusione: 71	Conclusione: 71
Penalty: 0	Penalty: 0

FILE NONE

Pianificazione EDF:	Pianificazione HPF:
9:0	6:0
10:9	7:7
6:17	10:10
7:24	2:18
2:27	4:28
1:37	9:38
8:43	1:47
4:52	8:53
5:62	5:62
3:67	3:67
Conclusione: 71	Conclusione: 71
Penalty: 312	Penalty: 228