



UNIVR Dipartimento di Informatica

Elaborato Assembly
Laboratorio Architettura degli Elaboratori

Pianificatore attività di un sistema produttivo

A.A. 2023/2024

Gruppo di lavoro:
Broccolato Marco (VR501013)
Drago Matteo (VR500241)

Specifiche

Si sviluppi **in Assembly (sintassi At&t)** un software per la pianificazione delle attività di un sistema produttivo, per i successivi 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: 17; Scadenza: 32; Priorità: 5;

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.

Variabili utilizzate

totaleRighe: quantità di righe lette dal file "Ordini.txt" che indicano la quantità di prodotti da ordinare;

file_scrittura: indirizzo del file su cui scrivere gli output;

fileScrittura: variabile che indica se è necessario scrivere su un altro file gli output o stamparli solo a video;

new_line_char: contiene il valore \n codificato in ASCII;

indirizzoRet: contiene l'indirizzo della return;

contatore: variabile per identificare il "campo" del prodotto di cui si sta controllando il range dei valori;

1 -> id, 2 -> durata, 3 -> scadenza, 4 -> priorità

numero: tag usato per contenere il numero durante il passaggio da stringa ad intero;

buffer: spazio per il buffer di input;

lines / totaleRighe: numero di \n letti dal file (quindi quantità di prodotti da ordinare);

posizione: salva la posizione dell'id del prodotto da scambiare per l'ordinamento;

Cima: salva la posizione della cima della pila (senza considerare la posizione della return);

check: indica se è già stato trovato un prodotto da scambiare;

numtmp: stringa temporanea;

numstr: stringa di output;

tempo: tempo di inizio produzione del prodotto;

tempoPenalita: tempo totale utilizzato per calcolare la penalità dei prodotti (tempo di fine produzione del prodotto);

totalePenalita: somma delle penalità dei singoli prodotti.

File/funzioni utilizzate

Il codice è suddiviso in 7 file:

pianificatore.s: file principale del programma, contiene tutti i collegamenti alle altre funzioni, il lancio dei 2 algoritmi di ordinamento e il controllo per la stampa su file di output;

intestazione.s: file che stampa a video l'interfaccia della scelta della pianificazione dei prodotti;

nome.s: file che si occupa di:

- prelevare l'indirizzo del file da leggere ed eventualmente su cui scrivere i risultati;
- convertire i numeri presenti nel file (come stringhe) in numeri interi;
- controllare il range di questi numeri;
- caricare i numeri nella pila.

edf.s e **hpf.s**: file che si occupano dell'ordinamento dei prodotti nella pila;

printOutput.s: file che si occupa della stampa dei risultati sul terminale;

printOutputFile.s: file che si occupa della stampa dei risultati sul file di output.

Logica del programma

Ordinamento EDF e HPF:

L'ordinamento di EDF (e HPF ma considerando altri parametri) si basa su un algoritmo di tipo selection sort.

All'interno della pila, in cui abbiamo precedentemente caricato i valori letti dal file di input, l'algoritmo ricerca il prodotto con la scadenza minore ed eventualmente, se 2 prodotti dovessero avere la stessa scadenza, ricerca quello con la priorità maggiore (nel caso di HPF, viene ricercato il prodotto con la priorità maggiore e, se ce ne fossero due uguali, quello con la scadenza minore). Fatto ciò, viene scambiata la posizione del prodotto selezionato con quello presente alla base della pila, questo fino a quando non ci saranno più prodotti da ordinare.

Lettura file:

La lettura del file consiste in una lettura carattere per carattere del file di input. Il programma costruisce così un numero intero, fino a quando non viene letto un carattere di andata a capo "\n". Dopo aver effettuato controlli sul range del numero letto, quest'ultimo viene inserito nella pila. Quando viene letto un carattere ",", il programma lo sostituisce con un carattere di andata a capo "\n".

121,10,23,3\n --> 121\n10\n23\n3\n

Stampa degli output su terminale:

Il programma seleziona le statistiche dei prodotti ordinati e stampa ordinatamente: l'id, il tempo di inizio produzione del prodotto, stringa "Conclusione", il tempo di fine produzione, stringa "Penalità", calcolo penalità.

Per il calcolo della penalità sfruttiamo la variabile tempoPenalita che contiene al suo interno la somma delle durate dei prodotti analizzati. Verrebbe effettuata la sua differenza con la scadenza del prodotto attualmente analizzato. Se la produzione del prodotto fosse in ritardo, viene calcolata la penalità.

File di Verifica

EDF.txt: penalità uguale a zero con EDF e maggiore di zero con HPF;

None.txt: penalità maggiore di zero con entrambi gli algoritmi;

Both.txt: penalità uguale a zero con entrambi gli algoritmi;

Minore10.txt: contiene un numero di prodotti inferiore a 10;

Maggiore10.txt: contiene un numero di prodotti maggiore a 10;

Error.txt: il “campo” di un prodotto non rispetta il range prestabilito (id del sesto prodotto).