

```
enum semafori,  
typedef struct{}msg  
typedef struct{}shm_mat
```

scan_child

starter_child

Sistemi Operativi
prova di laboratorio
– 26 luglio 2021 –

Creare un programma **my-du-s.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

my-du-s [path-1] [path-2] [...]

Il programma deve sostanzialmente simulare il comportamento del comando **du** con l'opzione **-s**: questo, per ogni percorso indicato, calcola lo spazio su disco occupato dai file in esso contenuti **ricorsivamente**.

Il processo padre, al suo avvio, creerà un unico processo figlio **Stater** ed un gruppo di processi **Scanner** (un processo distinto per ogni percorso radice indicato sulla riga di comando). I processi **Scanner** dovranno agire in parallelo e comunicheranno con il processo **Stater** unicamente attraverso un **segmento di memoria condiviso** con l'aiuto di un certo numero (minimo) di **semafori** da determinare a cura dello studente. Il processo **Stater** e il processo padre comunicheranno attraverso **una coda di messaggi**.

Ogni figlio **Scanner** provvederà ad effettuare la scansione ricorsiva del percorso radice a lui assegnato: per ogni file regolare incontrato invierà il suo pathname identificativo al processo **Stater**. Il processo **Stater**, per ogni richiesta ricevuta, determinerà lo spazio su disco occupato dal file utilizzando il campo **st_blocks** del record riportato dalla chiamata **stat** (vedere **man 2 fstat**). Tale informazione dovrà essere mandata al padre che conteggerà l'occupazione totale, distinta per ogni percorso radice, riportando alla fine sul suo standard-output tutti i totali (come il comando **du -s**).

Tutti i processi dovranno spontaneamente terminare alla fine dei lavori. Tutte le strutture persistenti di IPC dovranno essere correttamente rilasciate in uscita.

Un esempio di utilizzo potrebbe essere il seguente:

```
$ my-du-s /usr/share/dict /usr/share/pixmaps /etc/default  
122  /usr/share/dict  
1856 /usr/share/pixmaps  
56   /etc/default
```

Note:

- assumere una dimensione massima per il percorso più lungo da gestire: **MAX_PATH_LEN**;
- sarà necessario l'utilizzo di qualche meccanismo di segnalazione per la corretta terminazione spontanea del processo **Stater** e del processo padre principale.

Tempo: 2 ore e 20 minuti