

# [OS Internals] LAB 2.2 - QUESITI

*Esempi domande d'esame*

## Quesito #1

Descrivere tramite pseudocodice l'implementazione di `ram_stealmem(npages)` in OS/161 e motivare perché questa effettua allocazione di memoria contigua.

Una pseudo-implementazione di `ram_stealmem(npages)` è la seguente:

```
function ram_stealmem(npages): ptr addr
    size = npages * DIM_SINGOLA_PAGINA;
    if (primo_ind_libero+size>last_available_address)
        return 0;
    addr=primo_ind_libero;
    primo_ind_libero=addr+size;
    return addr;
end
```

L'allocazione di memoria che risulta dall'utilizzo (lato utente e lato kernel) di tale funzione è contigua perché ad ogni chiamata di `ram_stealmem()` il puntatore alla memoria disponibile (primo indirizzo utilizzabile) viene aumentato di una dimensione pari allo spazio richiesto per `npages`, in questo modo, finita la porzione di memoria messa a disposizione per un processo, comincia quella che occuperà (potenzialmente) un altro processo.

Inoltre con il solo utilizzo di `ram_stealmem()` - senza gestire quindi il rilascio dello spazio allocato - il SO procederà ad allocare finché c'è spazio disponibile fino a lanciare un'eccezione di `Out of memory`.

## Quesito #2

Motivare l'utilizzo e la differenza tra indirizzi fisici e virtuali nella gestione della memoria di OS/161. Fornire almeno un esempio per entrambi i casi.

L'uso di indirizzi fisici e virtuali, in OS161 come in tutti i sistemi operativi che fanno uso della memoria virtuale, vengono utilizzati affinché si possa avere a disposizione uno spazio di indirizzamento che è molto maggiore rispetto a quello fisico. Un **indirizzo fisico** è associato ad una locazione di memoria 'reale', un **indirizzo virtuale** (o **logico**) è un riferimento simbolico che sta ad indicare invece una locazione di memoria virtuale. Il programmatore, grazie a questo tipo di indirizzi, può scrivere dei programmi che siano indipendenti dalla memoria fisica realmente presente nel sistema. Volendo fornire degli esempi nel caso di OS161:

- **Indirizzo fisico** è ad esempio quello che indica l'indirizzo di base del segmento `KSEG0` del kernel, indicato con la costante `MIPS_KSEG0`; il tipo associato a tale categoria di indirizzi è `ptrdiff_t` (sta per *physical address type*).
- **Indirizzo virtuale** è ad esempio il parametro `addr`, passato alla funzione `free_kpages()`; il tipo associato è in questo caso `vaddr_t` (sta per *virtual address type*).

## Quesito #3

Descrivere il flow e le chiamate a funzione necessarie a liberare l'address space di un processo X che termina e allocare un nuovo address space di un processo Y

1. `as_destroy` → `free_ppages()`

- 2.