
PROGRAMIRANJE II
(05.10.2022)

- ❶ (20 bodova) Napisati funkciju čiji je prototip:

char* s raz(int (*f)(int), int n, ...);

Funkcija kao neobavezne argumente prima brojeve, pri čemu n predstavlja broj tih brojeva. Funkcija treba da sve brojeve spoji u jedan string, ali pri tom da primijeni transformaciju nad svakim od brojeva funkcijom f . Funkcija vraća dinamički alociran string koji sadrži sve brojeve spojene u jedan string koji su nastali nakon transformacije originalnih brojeva.

U glavnom programu demonstrirati upotrebu funkcije, pri čemu je potrebno kreirati minimalno tri neobavezna argumenta, te nakon poziva funkcije ispisati novokreirani string.

- ❷ (20 bodova) Definirati tip **PROCESS** kojim se reprezentuju podaci o jednom procesu čiji su atributi naziv (string od najviše 63 znaka) i vrijeme izvršavanja (u sekundama).

Napisati funkciju koja iz ulazne binarne datoteke, čiji je naziv prvi parametar funkcije, učitava podatke o procesima te formira i vraća uređen (sortiran) dinamički niz podataka o procesima. Podatke o procesima treba, jedan po jedan, učitavati iz datoteke i dodavati u dinamički niz, ali tako da su učitani podaci u nizu uvijek uređeni (*insert sort*). Podaci o procesima treba da budu sortirani po nazivu procesa. Preko parametra n treba vratiti ukupan broj učitanih podataka o procesima. Prototip funkcije je:

PROCESS* read(const char *file, int *n);

Napisati rekurzivnu funkciju koja vrši binarno pretraživanje niza od n podataka o procesima. Ključ pretrage je naziv procesa. Funkcija vraća adresu pronađenog podatka u nizu ili NULL ako traženi podatak ne postoji u nizu. Prototip funkcije je:

**PROCESS* find(PROCESS *arr, int n,
const char *name);**

- ❸ (20 bodova) Definirati tip(ove) kojim se reprezentuje linearna struktura podataka o procesima (podaci o jednom procesu su reprezentovani tipom **PROCESS** koji je definisan u 2. zadatku) sa *FIFO* disciplinom pristupa (ulančana reprezentacija). Definirati funkciju *enqueue* za dodavanje procesa u takvu strukturu, kao i funkciju *dequeue* za preuzimanje procesa iz takve strukture. Definirati i funkciju (korištenjem funkcija *enqueue* i *dequeue*) koja prihvata jednu takvu strukturu, a simulira izvršavanje procesa tako što se informacije o svakom procesu ispisuju na standardni izlaz.

- ❹ (20 bodova) Neka je definisan tip:

typedef struct cvor {

int info;

struct cvor *lijevi, *desni;

} CVOR;

kojim se reprezentuje čvor stabla binarnog pretraživanja.

Definirati funkciju koja na osnovu originalnog stabla kreira novo stablo binarnog pretraživanja koje sadrži one čvorove originalnog stabla čiji je informacioni sadržaj veći od parametra n . Koriijen originalnog stabla je argument funkcije. Funkcija kao rezultat vraća koriijen novog stabla. Prototip funkcije je:

CVOR* kopirajStablo(CVOR *cvor, int n);

U glavnom programu demonstrirati upotrebu funkcije, pri čemu je potrebno kreirati binarno stablo čija je dubina minimalno tri i čiji je koriijen parametar funkcije *cvor*, te ispisati u postorder redoslijedu novodobijeno stablo.